

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
2 October 2003 (02.10.2003)

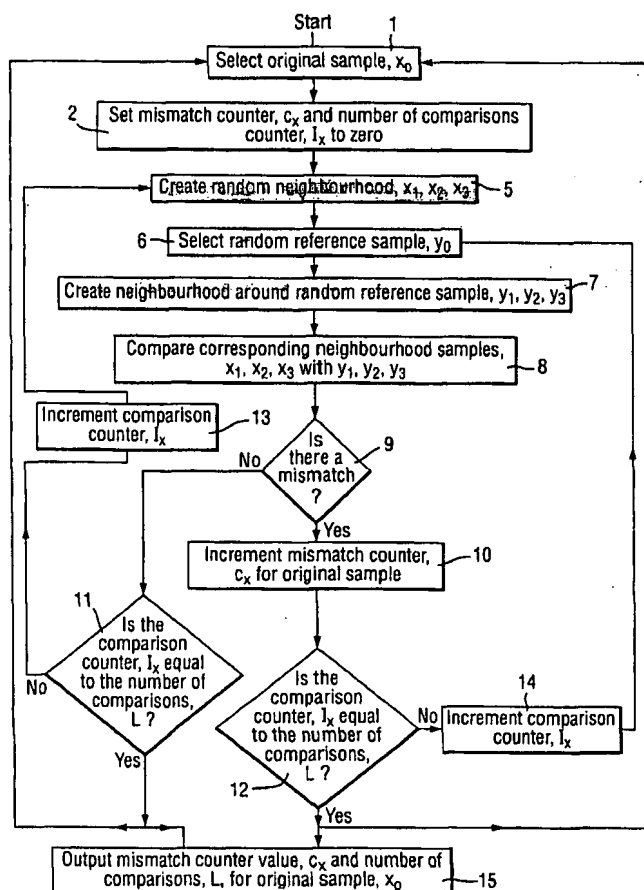
PCT

(10) International Publication Number
WO 03/081577 A1

- (51) International Patent Classification⁷: **G10L 19/00, H03M 7/30**
- (21) International Application Number: **PCT/GB03/01211**
- (22) International Filing Date: **24 March 2003 (24.03.2003)**
- (25) Filing Language: **English**
- (26) Publication Language: **English**
- (30) Priority Data:
0206851.8 22 March 2002 (22.03.2002) GB
0206853.4 22 March 2002 (22.03.2002) GB
0206854.2 22 March 2002 (22.03.2002) GB
0206857.5 22 March 2002 (22.03.2002) GB
- (71) Applicant (for all designated States except US): **BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).**
- (72) Inventor; and
(75) Inventor/Applicant (for US only): **STENTIFORD, Frederick, Warwick, Michael [GB/GB]; Sheepstor, Boyton, Woodbridge, Suffolk IP12 3LH (GB).**
- (74) Agent: **LLOYD, Barry, George, William; BT Group Legal Intellectual Property Department, Holborn Centre, 8th Floor, 120 Holborn, London EC1N 2TE (GB).**
- (81) Designated States (national): **AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.**
- (84) Designated States (regional): **ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),**

[Continued on next page]

(54) Title: **ANOMALY RECOGNITION METHOD FOR DATA STREAMS**



(57) Abstract: This invention identifies anomalies in a data stream, without prior training, by measuring the difficulty in finding similarities between neighbourhoods in the ordered sequence of elements. Data elements in an area that is similar to much of the rest of the scene score low mismatches. On the other hand a region that possesses many dissimilarities with other parts of the ordered sequence will attract a high score of mismatches. The invention makes use of a trial and error process to find dissimilarities between parts of the data stream and does not require prior knowledge of the nature of the anomalies that may be present. The method avoids the use of processing dependencies between data elements and is capable of a straightforward parallel implementation for each data element. The invention is of application in searching for anomalous patterns in data streams, which include audio signals, health screening and geographical data. A method of error correction is also described.

WO 03/081577 A1



European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *with international search report*

BEST AVAILABLE COPY

PCT

REC'D 23 JUN 2003

WIPO PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference A30114 WO	FOR FURTHER ACTION see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.	
International application No. PCT/GB 03/ 01211	International filing date (day/month/year) 24/03/2003	(Earliest) Priority Date (day/month/year) 22/03/2002
Applicant BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY		

This International Search Report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This International Search Report consists of a total of 4 sheets.

☒ It is also accompanied by a copy of each prior art document cited in this report.

1. Basis of the report

- a. With regard to the **language**, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

☐ the international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

- b. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international search was carried out on the basis of the sequence listing :

☐ contained in the international application in written form.

☐ filed together with the international application in computer readable form.

☐ furnished subsequently to this Authority in written form.

☐ furnished subsequently to this Authority in computer readable form.

☐ the statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

☐ the statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished

2. ☐ **Certain claims were found unsearchable** (See Box I).

3. ☐ **Unity of invention is lacking** (see Box II).

4. With regard to the **title**,

☐ the text is approved as submitted by the applicant.

☒ the text has been established by this Authority to read as follows:

ANOMALY RECOGNITION METHOD FOR DATA STREAMS

5. With regard to the **abstract**,

☒ the text is approved as submitted by the applicant.

☐ the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. The figure of the **drawings** to be published with the abstract is Figure No.

☒ as suggested by the applicant.

☐ because the applicant failed to suggest a figure.

☐ because this figure better characterizes the invention.

1
☐ None of the figures.

Anomaly Recognition

This invention relates to a system for recognising anomalies contained within a set of data derived from an analogue waveform, particularly, though not exclusively, for locating noise in an audio signal. The invention may be applied to data from many different sources, for example, in the medical field to monitor signals from a cardiogram or encephalogram. It also has application in the field of monitoring machine performance, such as engine noise. A noise removal system is also described for use in combination with the present invention.

Audio signals may be subject to two principal sources of noise: impulse noise and continuous noise.

There are a number of existing techniques for dealing with both sorts of noise. In particular, in order to reduce the effects of continuous noise, such as a background "hum" in audio data, low-pass filters, dynamic filters, expanders and spectral subtraction are used. However, these techniques suffer from the disadvantage that the characteristic of the noise must be known at all times. The nature of noise makes it impossible to perfectly characterise it. Thus, in practice, even the most sophisticated filters remove genuine signal that is masked by the noise, as a result of the noise being imperfectly characterised. Using these techniques noise can only be removed with any degree of success from signals, such as speech signals, where the original signal is known.

Impulsive noise, such as clicks and crackles, is even more difficult to process because it cannot be characterised using dynamic, time resolved techniques. There are techniques for correcting the signal. However, problems remain in identifying the noise in the first place. Most impulsive noise removal techniques assume that the noise can be detected by simple measurements such as an amplitude threshold. However, noise is in general unpredictable and can never be identified in all cases by the measurement of a fixed set of features. It is extremely difficult to characterise noise, especially impulsive noise. If the noise is not fingerprinted accurately all attempts at spectral subtraction do not produce satisfactory results, due to unwanted effects. Even if the noise spectrum is described precisely, the

results are dull due in part because the spectrum is only accurate at the moment of measurement.

Known impulse noise removal techniques include attenuation, sample and hold, linear interpolation and signal modelling. Signal modelling, as for example described in "Cedaraudio", Chandra C, et al, "An efficient method for the removal of impulse noise from speech and audio signals", Proc. IEEE International Symposium on Circuits and Systems, Monterey, CA, June 1998, pp 206-209, endeavours to replace the corrupted samples with new samples derived from analysis of adjacent signal regions. In this particular prior art technique, the correction of impulsive noise is attempted by constructing a model of the underlying resonant signal and replacing the noise by synthesised interpolation. However, notwithstanding the need to accurately detect the noise in the first place, this approach only works in those cases in which the model suits the desired signal and does not itself generate obtrusive artefacts.

The present invention provides a solution to the problems identified above with respect to noise identification and removal in data derived from an analogue waveform, in particular in audio signals. We have found that a technique developed, and described in our copending application EP-A-1 126 411, for locating anomalies in images, can be applied to data streams, in particular to audio signals. Our copending application describes a system which is able to analyse an image (2-D data) and highlight the regions that will 'stand out' to a human viewer and hence is able to simulate the perception of a human eye looking at objects.

Aspects of the present invention are provided as specified in the appended claims.

The first method of the invention allow for anomaly recognition in a data sequence, which is independent of the particular anomaly. As a specific example, this method will identify noise in a data sequence irrespective of the characteristics of the noise.

The present invention provides the advantages that it is not necessary for the signal or the anomaly to be characterised for the invention to work. An anomaly is

identified by its distinctiveness against an acceptable background rather than through the measurement of specific features. By measuring levels of auditory attention, an anomaly can be detected. Further, the invention does not rely upon specific features and is not limited in the forms of anomalies that can be detected.

- 5 The problem of characterising the anomaly is not encountered using the present invention.

Further, the invention does not rely upon specific features and is not limited in the forms of noise that can be detected. The problem of characterising the noise is not
10 encountered using the present invention.

One method includes the further steps of : identifying ones of said positional relationships which give rise to a number of consecutive mismatches which exceeds a threshold, storing a definition of each such identified relationship, utilising the
15 stored definitions for the processing of further data, and, replacing said identified ones with data which falls within the threshold. Having accurately identified the noise segment on the basis of its attention score, this method ensures that the noise is replaced by segments of signal that possess low scores and hence reduces the level of auditor attention in that region. Thus, in contrast to prior art techniques,
20 such as "Cedaraudio", this preferred method does not require any signal modelling.

This apparatus of the invention is preferably embodied in a general purpose computer, suitably programmed.

- 25 The invention also extends to a computer programmed to perform the methods of the invention, and to a computer program product directly loadable into the internal memory of a digital computer, comprising software code portions for performing the steps of the method of the invention, when said product is run on a computer.

- 30 This method allows for anomaly recognition in a data array, which is independent of the particular anomaly. As a specific example, this method will identify an anomaly in a data array irrespective of the characteristics of the noise.

In order that the invention may be more fully understood embodiments thereof will now be described by way of example only, with reference to the figures, in which

Figure 1 is a flowchart which illustrates schematically the operation of an embodiment of the invention;

Figure 2 is a flowchart which illustrates schematically the operation of a further embodiment of the invention;

Figure 3 is a flowchart which illustrates schematically the operation of a yet further embodiment of the invention;

Figure 4 illustrates schematically the basic components of a general purpose computer capable of performing the invention;

Figure 5 shows an example of a comparison between original sample, x_0 and random reference sample, y_0 ;

Figure 8 shows an example of the "hill climbing" embodiment of the present invention;

Figure 6 shows the failure of a static threshold;

Figure 7 shows a static threshold vs a dynamic threshold;

Figure 9 shows Result 1;

Figure 10 shows Result 2;

Figure 11 shows Result 3;

Figure 12 shows Result 4;

Figure 13 shows Result 5;

Figure 14 shows Result 6;

Figure 15 shows Result 7;

Figure 16 shows an example of how the error correction algorithm identifies a high anomaly score region;

Figure 17 shows an example of how the error correction algorithm creates counters;

Figure 18 shows an example of how the error correction algorithm carries out the comparison and logging process;

Figure 19 shows an example of how the error correction algorithm moves a neighbourhood during error correction;

Figure 20 is a flow chart depicting the steps of shape learning error correction;

Figure 21 shows Result 8.

5 Figure 22 is a flowchart which illustrates schematically the operation of an embodiment of the invention;

Figure 23 illustrates schematically the basic components of a general purpose computer capable of performing the invention;

Figure 24 shows an example of a waveform with cycles;

10 Figure 25 shows area definitions of the cycles;

Figure 26 shows an example of padding a cycle;

Figure 27 shows the Measure of Difference using a first denominator – the Larger Area Of Two Cycles;

15 Figure 28 shows the Measure of Difference using a second denominator - $|MaxArea - MinArea|$

Figure 29 shows Result 1a;

Figure 30 shows Result 2a;

Figure 31 shows Result 3a;

Figure 32 shows Result 4a;

20 Figure 33 shows Result 5a;

Figure 34 shows Result 6a;

Figure 35 shows Result 7a;

Figure 36 shows Result 8a;

Figure 37 shows Result 9a;

25 Figure 38 shows Result 10a;

Figure 39 shows cutting erroneous cycles;

Figure 40 shows replacing erroneous cycles.

30 The ordered sequence of elements which form the data is represented in an array derived from an analogue waveform. Although the data may be a function of more than one variable, in this invention the data is "viewed" or ordered in dependence on one variable. Thus, the data can be stored as an array. The array is a one

dimensional array, a $1 \times n$ matrix. Data in a one dimensional array is also referred hereinbelow as one dimensional data. The values of the data contained in the array may be a sequence of binary values, such as an array of digital samples of an audio signal. One example of the anomaly recognition procedure is described below in connection with Figures 1-8, where the neighbouring elements of x_0 are selected to be within some one-dimensional, distance of x_0 . (Distance between two elements or sample points in this example may be the number of elements between these points).

10 Detection of anomalies in data represented in a one-dimensional array (eg: time resolved data or audio data or data from an acoustic source) concerns instructing a computer to identify and detect irregularities in the array in which the set of data is arranged. There are various reasons why a particular region can be considered as 'irregular' or 'odd'. It could be due to its odd shape or values when compared with
15 the population data (the remainder of the data); it could be due to misplacement of a certain pattern in a set of ordered pattern. Put more simply, an anomaly or irregularity, is any region which is considered different to the rest of the data due to its low occurrence within the data: that is, anomalous data will have one or more characteristics which are not the same as those of the majority of the data.

20

In the specific examples given in the description of the invention, the algorithm is tested mainly on audio data with the discrete samples as the one-dimensional data. However, the invention is limited in no way to audio data and may include other data that can be represented in a one dimensional array derived from a waveform
25 having a plurality of cycles.

The software which, when run on a computer implements the present invention, "One Dimensional Anomaly Detector", is written in Curl language using Curl Surge Lab IDE beta 5 – Build: 1.6.0 release/englewood/O-1237: copyright© 1998-2001
30 and may not be compatible with future releases of Curl. The results shown in this description were produced by the software mentioned above. Again, however, the

invention is not limited to software written using this particular language and may be implemented using other computer languages.

5 This algorithm of the present invention works on the basis of analysing samples. A further algorithm described later as the "cycle comparison algorithm" compares cycles defined by certain zero crossings.

The method for the sample analysis algorithm will now be described with reference to Figures 1 to 8.

10

The components shown in Figure 4 include a data source 20 and a signal processor 21 for processing the data. The data is either generated or pre-processed using Cool Edit Pro – version 1.2: Cool Edit Pro is copyrighted © 1997-1998 by Syntrillium software Corporation. Portions of Cool Edit Pro are copyrighted © 1997, 15 Massachusetts Institute of Technology. The invention is not limited in this respect, however, and is suitable for data generated or preprocessed using other techniques. Figure 4 also shows a normaliser 22. The data is normalised by dividing all values by the maximum modulus value of the data so that the possible values of the data range from -1 to 1.

20 A central processing unit (CPU) 24, an output unit 27 such as a visual display unit (VDU) or printer, a memory 25 and a calculation processor 26. The memory 25 includes stores 250, 254-256, registers 251, 257-259 and a mismatch counter 253 and a comparison counter 252. The data and the programs for controlling the computer are stored in the memory 25. The CPU 24 controls the functioning of the 25 computer using this information.

With further reference to Figures 1-5, a data stream to be analysed is received at the input means 23 and stored in a digital form in a data store 250, as a one dimensional array, where each datum or data element has a value attributed to it. 30 An original sample of data, x_0 , (a reference test element) is selected (step 1) from the one dimensional array, and its value is stored in an original sample register 251.

A mismatch count, cx , stored in a mismatch counter 253, and a count of the number of data comparisons, lx , stored in a comparison counter 252, are both set to zero (step 2).

5 Then a random neighbourhood, x_1 , x_2 , x_3 , (test elements) which comprises a number of data in the vicinity of the original sample (reference test element), x_0 , of a certain size (PARAMETER: neighbourhood size) is selected from neighbouring samples (step 5). The neighbourhood is chosen to lie within a particular range (or "neighbourhood range") (PARAMETER: radius) from the original sample, x_0 .

10 Then, a second reference sample, y_0 , is randomly chosen anywhere within a certain domain or range (PARAMETER: comparison domain) in the set of data (step 6). The neighbourhood, (i.e. test elements) x_1 , x_2 , x_3 selected around the original sample, x_0 together with the original sample, x_0 , have a certain configuration which makes a 'pattern'.

15 The neighbourhood, y_1 , y_2 , y_3 , (comparison elements) selected around the random reference sample, (the reference comparison element) y_0 , together with the reference sample, y_0 , are chosen to have the same configuration, or pattern, as the neighbourhood around the original sample.

20 In the embodiments shown in Figures 1 and 3A and 3B, the values of the data in the original sample 'pattern' (test group), x_0 , x_1 , x_2 , x_3 are then compared by calculation processor 26, with the values of the data in the reference sample 'pattern' (comparison group), y_0 , y_1 , y_2 , y_3 , defined by the reference sample together with its neighbouring samples (step 8). If the absolute value of the difference, $|x_0 - y_0|$, $|x_1 - y_1|$, etc, between two respective samples or elements is
25 more than a certain threshold (PARAMETER: threshold), then it is considered as being 'different'. If one or more samples in the original sample pattern are different from the reference sample pattern, then it is said that a mismatch occurred. The choice of the threshold can optionally be varied, and may depend on the range of values within the set of data. In the embodiment shown in Figure 2, this part of the
30 algorithm is carried out according to similar principles but different values are compared. This is described below in more detail with reference to Figures 2 and 6.

In all other respects, however, the algorithm shown in Figure 2 is the same as those shown in Figures 1 and 3A and 3B.

- Further, with reference to Figures 1-5, when a mismatch occurs, the mismatch counter, cx , for the original sample, $x0$, is incremented (step 10). In this case the neighbourhood (test group) around the original sample (reference test element) is kept, i.e., the original sample pattern is kept, and the program returns to step 6 to choose another random 2nd reference sample, $y0$, for the same comparison process.
- 10 When a match occurs the mismatch counter, cx , is not increased. The program returns to step 5 which creates a new neighbourhood around the original sample, whose configuration has a new pattern, before moving on to choose another random 2nd reference sample (step 7) for the comparison step (step 8).
- 15 For each original sample, $x0$, a certain number of comparisons, L , are made which result in a certain number of mismatches and matches. The total number of mismatches plus matches is equal to the number of comparisons (step 11 and step 14). The number of comparisons can be varied and will depend on the data to be analysed and the processing power available. Also, the greater the number of
- 20 comparisons, the greater the accuracy of the anomaly detection.

Once the comparison step (step 8) has been done the certain number of times, L , the program returns to step 1 to select a different original sample, $x0$ and the mismatch counter value, cx , and the number of comparisons, L , is output for original sample, $x0$ (step 15).

Whether the original sample or reference test element, $x0$, is judged to be an anomaly will depend on the number of mismatches in comparison to the number of comparisons, L . The normalised anomaly scores for each original sample, $x0$, are obtained by dividing the mismatch counter, cx , for each sample, $x0$, by the number of comparisons, L , which is also equal to the maximum mismatch count, so that the

anomaly score ranges from zero to one, with zero being 0% mismatch and one being maximum mismatch.

Figure 5 shows an example of a one-dimensional data with each box representing a sample. Sample marked 'x' is the original sample and sample marked 'y' is the randomly chosen reference sample. The samples, x1, x2, x3, are the neighbourhood samples whose configuration make up the original sample pattern. In the example shown in Figure 5, the radius (or neighbourhood range) is equal to 3, the neighbourhood size is equal to 3 and the comparison domain is equal to the region where y is chosen. A mismatch occurs if $|x_n - y_n| > \text{threshold}$, where, n, the neighbourhood size takes a value from 1 to 3.

As shown in Figure 5, the first sample which could be scored is the sample with a distance 'radius' away from the start and the last sample to be scored is the sample with a distance 'radius' away from the end.

By way of further explanation of the above example of comparison, a numerical example is set out in Table 1.

Sample Index, n	(normalized) Value of X_n	(normalized) Value of Y_n	Threshold value	Value of $ Y_n - X_n $	Mismatch ?
0	0.75	0.70	0.2	0.05	No
1	-0.90	-0.71	0.2	0.19	No
2	0.01	0.34	0.2	0.33	YES
3	0.23	0.45	0.2	0.22	YES

Table 1: Example of Comparison

In the examples given, two of the samples mismatch. As long as one or more samples in the neighbourhood mismatches, the mismatch counter for the original, in this example, X_0 , will be incremented by one.

With reference to Figures 2 and 6, the inventor has noticed that when the waveform becomes complex or the sampling rate is increased the number of mismatches increases relative to the number of matches. This causes the scores to become saturated. As the complexity of the waveform increases the probability of

picking a random reference Y sample that matches the original sample X decreases. Similarly, as the sampling rate is increased, the probability of finding a match decreases. The increased probability of having a mismatch causes saturation of the scores.

5

To alleviate the problem of score saturation, a 'hill climbing' strategy has been developed to improve the likelihood of a match. The strategy is called "hill climbing" because when a mismatch is found, the waveform is "climbed" in both directions along the ordered set of data elements until a match is found.

10

Figure 2 is a flow diagram showing the steps an algorithm including the "hill climbing" process and how they fit in with the steps of the sample analysis algorithm described above. The hill climbing process is shown within the dotted line 20. It is seen in Figure 2 that the 'hill climbing' process includes some additional steps to the sample analysis algorithm shown in Figure 1.

15

The "hill climbing" process is explained with reference to Figures 2 and 6. First the original sample, marked X, is chosen (step 1). The neighbourhood samples, coloured medium dark grey in Figure 8 and shown in the neighbourhood of the original sample X, are then selected either randomly (step 5) or reused from the previous comparison if a mismatch occurred previously (refer to step 6). In the example shown in Figure 8, the neighbourhood size (parameter: neighbourhood size) is three, hence three neighbouring samples are selected. And the furthest distance from which a neighbouring sample can be selected is the radius (parameter: radius), which is equal to four in the example in Figure 8. These samples make up the original "pattern" (step 5).

20

25

30

Next, a reference sample, marked Y, is randomly chosen from anywhere in the data within a certain domain (step 6) (parameter: comparison domain, not shown in Figure 8, but shown for example, in Figure 5).

Then the reference sample, Y, is compared with the original sample, X (step 22). It is determined whether there is a mismatch between the reference sample and original sample (step 24). In the example shown in Figure 8, the reference sample Y lies outside the threshold (parameter:threshold) region of the original sample X, hence it does not match the original sample X. Therefore, in the case of this mismatch the next step (steps 26, 28, 30 and 32) is to 'hill climb' the reference sample by searching the samples within a search radius around Y for a sample that matches with the original sample X. This searching is done one sample at a time in both directions along the one dimensional array (step 30).

10

In Figure 8 the sample marked A is the first sample near sample Y that matches the original sample X as it falls within the threshold region. Next, the neighbourhood samples of X (coloured medium dark grey) are compared with the corresponding neighbourhood samples of A (step 28). If they match (step 32), then the mismatch counter is not increased and the process is continued with the next comparison by selecting another random reference sample (step 6). In the example shown in Figure 8, the corresponding neighbourhood samples X and A do not match (step 32), but in spite of this and in contrast to the steps shown in Figure 1, the mismatch counter for sample X is not increased.

15

Instead of increasing the mismatch counter, the 'hill climbing' process is continued as described above. Eventually, sample marked B is selected and found to match the original sample X. Then the neighbourhood samples of X (coloured medium dark grey) are compared with the corresponding neighbourhood samples of B (step 28). If they match one another, then the next comparison is continued with by selecting another random reference sample (step 6). In the example shown in Figure 8 they do match, so the mismatch counter is not increased, and the process is continued with the next comparison by selecting another random reference sample.

20

25

30

It can be seen by reference to Figure 8 and the explanation above, the 'hill climbing' process stops when one of two things happen. The process stops when the algorithm finds a matching "pattern". Alternatively, the other way the 'hill climbing' process stops is when the algorithm fails to find any matching "pattern" within a

certain search radius for the 'hill climbing' (illustrated in Figure 8). The radius being set to be equal to the radius of the original sample X's neighbourhood (parameter:radius). The algorithm searches all samples within the search radius (step 26). When the algorithm fails to find any matching "pattern" in the neighbourhood, then the mismatch counter for original sample X is increased (step 5 10).

Therefore, the mismatch counter for the original sample only increases when there is no matching pattern within the 'hill climbing' search radius from the randomly selected reference sample. By only increasing the mismatch counter when 10 there are no matching "patterns" in the neighbourhood of the reference sample, the constraints imposed on the search for a match are relaxed. Thus, the probability of finding a match are increased. This process is successful in eliminating the problem of saturation of the scores observed by the inventors. Reference is made to Figures 15 10 to 15 which show the results achieved.

With reference to Figure 6, the inventor has found, that in addition to the problem of saturation another problem exists. Due to the effects of constant sampling rate, samples which lie on a larger gradient are more distant apart compared to samples 20 which lie on a small gradient.

This is because a constant sampling rate means samples are taken at equal intervals of time. When the waveform changes rapidly, i.e. has a large magnitude of gradient, the difference between two subsequent sample values is therefore large. When the 25 waveform has a small gradient, there is only a slight difference between two subsequent sample values. See Figure 6 for illustration.

The effect of a static threshold or mismatch criterion while comparing samples is as follows: samples which lie on the larger gradient will be discriminated and have high 30 mismatch scores as they are less likely to match with their neighbours. This will result in an artificially high mismatch score for data lying on a steep gradient. Similarly, data lying on a shallow gradient will score too low.

The inventor has found that this detrimental effect can be removed by using a dynamic threshold, which takes into account the local gradient of the samples. The dynamic threshold is an adaptive variable threshold that is dependent on the sample's local gradient.

5

The dynamic threshold may be defined as:

$$\text{DynamicThreshold} = \frac{\text{LocalGradient}}{2} + \text{StaticThreshold}$$

10 In sampling an analogue waveform (see Figure 2) discrete samples are taken over equal time intervals. Each sample acts as a representative for the particular interval. In this interval the waveform however assumes different values. The local gradient can be defined as the difference between the boundary values of the interval and is a measure of the variation in the interval (the intervals will be chosen smaller than
15 any periodicity of the waveform). In this way, the sample interval is set to have a non-dimensional value of 1. By defining a dynamic threshold which increases with increasing local gradient, for example by adding a term proportional to the gradient as above to a static threshold value, the mismatch criterion is increased for steeper gradients and sampled values may thus differ more before they mismatch. For small
20 gradients, samples are mismatched if they differ by a smaller threshold amount. The mismatch criterion or threshold is thus adaptive to the particular environment of a sample. (PARAMETER: threshold). The static threshold can be determined to suit the particular data and sensitivity required. Similarly, the particular form of the gradient responsive term may vary according to the sampled data and could be
25 determined empirically. (Obtaining a dynamic threshold is optional, and a static threshold is possible instead).

In Figure 7, the upper spectrum shows the result with striations due to discrimination on large slopes using the static threshold while the spectrum below
30 shows a more uniform attention score as a result of dynamic thresholding.

In the above example, the data comprises an analogue waveform which is sampled at regular intervals, although it will be appreciated that the intervals need not be regular.

- 5 Figure 2 shows the steps taken in the case where an analogue waveform is sampled, and includes the step 3 of determining the gradient at the original sample, x_0 , and step 4 of determining the dynamic threshold. In step 8, the corresponding neighbourhood samples are compared with the dynamic threshold.
- 10 Figure 3 shows the steps taken in the case of an array of digital data, and includes step 16 of determining the values of samples neighbouring the original sample, and step 17 determining the dynamic threshold. In step 8, as for the case of an analogue waveform, the corresponding neighbourhood samples are compared with the dynamic threshold.

15

The gradient determination step and the step of determining the values of samples neighbouring the original sample are carried out by the calculation processor 26, and the values determined are stored in the register 259, where they are accessible as the dynamic threshold value for use in the comparison step (step 8).

20

- Both the "hill climbing" process and the dynamic threshold process may be implemented independently to one another as shown in Figures 2, 3A and 3B. Alternatively, they may be implemented in combination with each other. In particular, the "hill climbing" process described above with reference to Figures 2 and 6 is suitable for combination with either of the dynamic threshold embodiments shown in Figures 3A and 3B.
- 25

- Figures 9 to 15 show Results 1 to 7, respectively. The results shown in these Figures are produced after the implementation to the sample analysis algorithm described with reference to Figures 1 and 5 of a combination of the "hill climbing" shown in Figures 2 and 6 and the dynamic threshold processes shown in Figures 3A and 3B described above.
- 30

The results show good anomaly discrimination with no saturation.

5 The comparison domain for these results is the entire data length. The results show in the lower part of the diagram the input data for analysis. The upper portion of the diagram shows the mismatch scores achieved for each sample using the sample analysis algorithm plus the "hill climbing" and dynamic threshold modifications. In the upper portion, an anomaly is identified as being those portions having the highest mismatch scores.

10

The results shown are for audio signals. However, the present invention may also be applied to any ordered set of data elements. The values of the data may be single values or may be multi-element values.

15 Result 1 shown in Figure 9 shows a data stream of 500 elements having a binary sequence of zeros and ones. The anomaly to be detected is a one bit error at both ends of the data. In this example, the number of comparisons was 500, the radius was equal to 5, the neighbourhood size was equal to 4 and the threshold was equal to zero. The peaks in the upper portion of the graph show a perfect discrimination
20 of the one bit errors at either end of the data array.

Result 2 shown in Figure 10 shows data stream having the form of a sine wave with a change in amplitude. In this example, the number of comparisons was 500. The radius was equal to 5, the neighbourhood size was equal to 4 and the threshold
25 was equal to 0.01. The peaks in the upper portion of the graph show a perfect discrimination of the anomaly. The highest mismatch scores being for those portions of the data stream where the rate of change of amplitude is the greatest.

Result 3 shown in Figure 11 shows a data stream having the form of a sine wave with background noise and burst and delay error. In this example, the number of
30 comparisons was 500, the neighbourhood size was equal to 4 and the threshold

was equal to 0.15. The peaks in the upper portion of the graph show a good discrimination of the anomalies present.

Result 4 shown in Figure 12 shows a data stream having the form of a 440kHz sine wave that has been clipped. The data has been sampled at a rate of 22kHz. In this example, the number of comparisons was 1000, the radius was equal to 75, the neighbourhood size was equal to 4 and the threshold was equal to 0.15. The peaks show a good discrimination of the anomalies. Further, it is commented that the gaps in between the peaks can be eliminated by selecting a larger neighbourhood size.

Result 5 shown in Figure 13 shows a data stream having the form of a 440kHz sine wave that has been clipped. The data has been sampled at a rate of 11kHz. In this example, the number of comparisons was 1000, the radius was equal to 10, the neighbourhood size was equal to 5 and the threshold was equal to 0.15. The peaks show a good discrimination of the anomalies.

Result 6 shown in Figure 14 shows a data stream having the form of a 440kHz sine wave including phase shifts. The data has been sampled at a rate of 44kHz. In this example, the number of comparisons was 1000, the radius was equal to 50, the neighbourhood size was equal to 4 and the threshold was equal to 0.1. The peaks show good discrimination of the anomalies.

Result 7 shown in Figure 15 shows a data stream having the form of a 440kHz sine wave including phase shifts. The data has been sampled at a rate of 44kHz. In this example, the number of comparisons was 1000, the radius was equal to 50, the neighbourhood size was equal to 4 and the threshold was equal to 0.1. The peaks show near perfect discrimination of the anomalies.

An error correction system is now described with reference to Figures 16-20, which has application to the present invention. Having used the anomaly detection system previously described to identify regions of anomaly in a waveform, error correction is provided to remove the detected errors. From the attention map

produced as described above, a suitable filter coefficient is set (PARAMETER: filter coefficient) so that only the anomalous region remains in the map before passing the data to an error correction algorithm.

5 The error correction algorithm used depends on the algorithm used to detect the anomaly. For example, a cycle comparison detection algorithm is described further below which is for use together with a cutting and replacing correction algorithm. It has been found that a shape learning error correction
10 in this application. The shape learning algorithm is described below.

 The shape learning error correction described below may be implemented directly. The success of the error correction however, is dependent primarily on
15 being able to pinpoint the anomaly with confidence, which is the function of the detection algorithm.

 The error correction method described below deals with the error by taking a closer look at what is happening when the detection algorithm does the
20 comparison described above. Figure 16 shows that due to the nature of the detection algorithm, the first and last samples in a high score region are not amongst the erroneous samples. The first sample and last sample that have high score are a distance of 'radius' (PARAMETER: radius) away from the first and last erroneous sample. This is because the first neighbourhood that may select the
25 erroneous sample as one of the neighbourhood samples normally lies a distance 'radius' away.

 To explain the details of how the algorithm works the example given in Figure 16 is referred to. A region of anomaly is indicated with high scores but the actual samples that are erroneous have lower scores than the indicated samples. The
30 algorithm does the error correction routine starting from the left-hand side towards the right-hand side. First, as shown in Figure 17, it takes the first sample from the

left with a high score and creates two counters for each sample within the radius of the first sample.

5 All samples, X_0 to X_6 , are then compared with other parts of the data. This comparison method is similar to the detection algorithm and uses the two parameters from the detection algorithm, which are the number of comparisons (PARAMETER: number of comparisons) and the static threshold value (PARAMETER: threshold). X is considered as the original sample. This comparison method uses the dynamic thresholding that is used in the detection algorithm
10 described above.

For each comparison of the neighbourhood, X_0 to X_6 , with other parts of the data, if the number of samples in the neighbourhood that mismatches is less than or equal to a value called 'range' then certain information will be logged in the counters for those samples that mismatch, refer to Figure 18. The value 'range' is given by the
15 parameter "proportion to fix at one go" (PARAMETER: proportion to fix at one go) multiplied by the 'radius' (PARAMETER: radius) rounded to the nearest integer. The parameter proportion to fix at one go can take a value between 0 and 1. Hence the value 'range' takes a minimum value of 1 and maximum value of 'radius'.

20 Two examples are given in Figure 18. When X and Y is compared, only one sample mismatches, which is less than the value of 'range'. So, the 'mismatch frequency' counter is increased for the sample(s) that mismatch and the 'total mismatch value' counter is also updated by adding it with the value of the difference between X and Y (the mismatch value). However, when X and Z is compared, four samples
25 mismatch, which is more than the value of 'range'. If this happens, no information is logged. The counter values are not altered.

At the end of the comparison process, the 'mismatch frequency' counter holds the value indicating how often each of the samples X_0 to X_6 mismatches, and the 'total mismatch value' counter holds the sum of all the mismatch difference values that
30 have occurred for each of the samples X_0 to X_6 . From these two pieces of information, we can now decide which sample(s) are always causing a mismatch and how much to adjust them so that they will match more often. This can be done

by first getting a mean value for the mismatch frequencies of all the samples. Then any sample(s) that have a larger mismatch frequency than the mean value will be considered needing adjustment. The amount to adjust each sample is given by the average value of the mismatch values. This average value is obtained by dividing
5 the value in the 'total mismatch value' counter by the value in the 'mismatch frequency' counter of the sample(s) that need to be adjusted.

The sample(s) are then adjusted and the new attention score for the sample X_0 is obtained using the standard detection algorithm. If the new attention score is less than the previous score, the adjustments are kept, otherwise the adjustments are
10 discarded. The algorithm repeats the process again for neighbourhood X_n and does the adjustments again as long as the attention score for X_0 decreases. If the attention score for X_0 does not decrease after a certain number of times (PARAMETER: number of tries to improve score) consecutively, the algorithm moves on the next sample to be chosen as the original sample. The next sample to
15 be chosen lies 'range' number of samples to the right of the previous original sample. Figure 19 illustrates how the algorithm uses the 'range' value as described above.

As shown in Figure 19, for each new step the algorithm takes, the new original
20 sample X_0 lies 'range' samples in front of the previous original sample. This also means that the new neighbourhood will contain 'range' number of erroneous samples, assuming that all the errors in the previous neighbourhood are corrected perfectly. Because of this, when the neighbourhood is compared to an identical reference neighbourhood elsewhere in the data, it is expected that only 'range'
25 samples to mismatch while the rest of the samples should match. If more than 'range' samples mismatch, this means that the good samples are also mismatching, hence the reference neighbourhood that it compared with is unlikely to be identical to the original neighbourhood and therefore no information at all is logged.

The algorithm is called shape learning because it tries to make adjustments to the
30 erroneous samples so that the overall shape or recurring pattern of the waveform is preserved. As the total number of samples is the same before and after the error correction, the algorithm works fine if the error is not best fixed by inserting or

removing samples. If this is the case, then the algorithm will propagate the error along the waveform. This is due to the error correction routine which starts from the left of the 'high score' region and adjusts the samples towards the right. Figure 21, Result 8 shows a good example of the phase shift error described above. In Figure 21, the lower part of the diagram shows the input data for analysis. The upper portion of the diagram shows the results of the analysis where the y axis in the upper portion shows the mismatch value. In the upper portion, an anomaly is identified as being those (lighter) portions having the greatest mismatch values. It is noted that Result 8 is shown to illustrate the phase shift. The error recognition has been achieved not using the algorithm described in this application, but using the cycle comparison algorithm described further below.

Figure 20 shows a flow chart outlining the steps of the shape learning error correction described above.

Firstly, the first "high score" original sample, X, and its neighbourhood are obtained, step 100. Next, counters are created for each of the samples in the neighbourhood, step 102. A random reference sample and its neighbourhood are also selected, step 104. Having done this, the entire neighbourhood is compared, step 106, and it is determined whether more than the "range" of samples mismatch. If the answer is "yes", the comparison counter is increased, step 114, and the algorithm returns to step 104 to select a random reference sample and its neighbourhood. If the answer is "no", the next step is to obtain the difference, the mismatch value, dn , for the sample or samples that mismatch, step 108. Then the mismatch frequency counter is increased and the mismatch value, dn is added to the mismatch value counter for the sample or samples that mismatch, step 110.

Next, it is determined whether the comparison counter is equal to the number of comparisons, step 112. If the answer is "no" the algorithm returns to step 114, and the comparison counter is increased before the algorithm returns to step 104 to select a random reference sample and its neighbourhood. If the answer is "yes", the mean of the mismatch frequency counters is obtained, step 116. Subsequently, the sample or samples whose mismatch frequency counter is more

than the calculated mean in step 116, are identified, step 118. The identified sample or samples are adjusted by their average mismatch value, step 120. Having done this, a new attention (mismatch) score is obtained for the original sample using the sample analysis detection algorithm described above, step 122. The new
5 attention (mismatch) score is compared with the old (first) attention score, step 124. If it is lower than the old score, the adjustments made are kept and the failed counter is reset. If the new score is not lower, the adjustments made are discarded and the failed counter is increase, step 126.

10 Next, it is determined whether the failed counter is equal to the number of tries to fix the error, step 130. If the answer is "no", the algorithm returns to step 104 to select a random reference sample and its neighbourhood. If the answer is "yes", the next original sample, X, and it neighbourhood is obtained, step 132, before the algorithm returns to step 102, to create counters for each of the samples
15 in the neighbourhood.

 Depending on the type of error and the original waveform, certain methods could prove to be more efficient in removing the error. The shape learning algorithm described above, requires large amounts of processing time due to its looping
20 construct. But nevertheless it is the preferred way of removing the error as it possesses the ability to predict the shape of the waveform. However, on occasion it propagates certain errors as it does not alter the total number of samples. Cutting or replacing as described in our copending unpublished application (IPD reference A30176) proves to be the best method in such cases. Further, it is noted that in
25 any case the performance of the error correction is dependent on the performance of the anomaly detection algorithm.

 A detection algorithm of the present invention has been demonstrated to be very tolerant to the type of input data as well as being very flexible in spotting
30 anomalies in one-dimensional data. Therefore there are many applications where such detection method may be useful.

In the audio field, such a detection algorithm may be used as a line monitor to monitor recordings and playback for unwanted noise as well as being able to remove it. It may also be useful in the medical field as an automatic monitor for signals from a cardiogram or encephalogram of a patient. Apart from monitoring human signals, it may also be used to monitor engine noise. Like monitoring in humans, the output from machines, be it acoustic signals or electrical signals, deviate from its normal operating pattern as the machine's operating conditions vary, and in particular, as the machine approaches failure.

The algorithm may also be applied to seismological or other geological data and data related to the operation of telecommunications systems, such as a log of accesses or attempted accesses to a firewall.

As the detection algorithm is able to give a much earlier warning in the case of systems that are in the process of failing, in addition to monitoring and removing errors, it may also be used as a predictor. This aspect has application for example, in monitoring and predicting traffic patterns.

A further embodiment, the referred to as the "cycle comparison" is now described.

Detection of anomalies in an ordered set of data concerns instructing a computer to identify and detect irregularities in the set. There are various reasons why a particular region can be considered as 'irregular' or 'odd'. It could be due to its odd shape or values when compared with the population data; it could be due to misplacement of a certain pattern in a set of ordered pattern. Put more simply, an anomaly or irregularity, is any region which is considered different due to its low occurrence within the data.

In the specific examples given in the description of the invention, the algorithms are tested mainly on sampled audio data with the discrete samples as the one-dimensional data. However, the invention is limited in no way to audio data and may include, as mentioned above other data, or generally data obtained from an acoustic source, such as engine noise or cardiogram data.

This algorithm of the present invention works on the basis of identifying and comparing cycles delimited by positive zero crossings that occur in the set of data. The inventors have found however, that the sample analysis algorithm as described above may start to fail when the input waveform becomes too complex. Although the 'hill climbing' method described above has been implemented, saturation is still occurs for more complex waveforms. Saturation is an effect observed by the inventors when waveforms become complex or the sampling rate is increased. In these circumstances, the number of mismatches increases relative to the number of matches without necessarily indicating an anomaly. As the complexity of the waveform increases the probability of picking a random reference Y sample that matches the original sample X decreases. Similarly, as the sampling rate is increased, the probability of finding a match decreases. The increased probability of having a mismatch causes saturation of the scores.

Also, using the "hill climbing" method the processing time required to analyse a 1s length of audio data sampled at 44kHz sampling rate uses a lot of processing time, requiring up to 220s of processing time on a PII 266MHz machine.

The method for the cycle comparison algorithm will now be described with reference to Figures 22 to 28.

The components shown in Figure 22 include a data source 20 and a signal processor 21 for processing the data, a normaliser 22 and an input 23. The data is either generated or pre-processed using Cool Edit Pro – version 1.2: Cool Edit Pro is copyrighted © 1997-1998 by Syntrillium software Corporation. Portions of Cool Edit Pro are copyrighted © 1997, Massachusetts Institute of Technology. The invention is not limited in this respect, however, and is suitable for data generated or preprocessed using other techniques.

Also shown in Figure 2 is a central processing unit (CPU) 24, an output unit 27 such as a visual display unit (VDU) or printer, a memory 25 and a calculation processor 26. The memory 25 includes stores 250, 254-256, registers 251, 257-259 and a mismatch counter 253 and a comparison counter 252. The data and the

programs for controlling the computer are stored in the memory 25. The CPU 24 controls the functioning of the computer using this information.

5 With reference to Figures 22-28 where indicated, a data stream to be analysed is received at the input means 23. Firstly, the data is normalised by normaliser 22 by dividing all values by the maximum value of the data so that the possible values of the data range from -1 to 1.

The normalised data is stored in a digital form in a data store 250, as a one dimensional array, where each datum has a value attributed to it.

10

Then the algorithm identifies all the positive zero crossings in the waveform (step O). A mean DC level adjustment (not shown) may also be made before the positive zero crossings are identified, to accommodate any unwanted DC biasing.

15 The positive zero crossings are those samples whose values are closest to zero and if a line were drawn between whose neighbours, the gradient of the line would be positive. For example, of the sequence of elements having the following values: -1, -0.5, 0.2, 0.8, 1, 0.7, 0.3, -0.2, -0.9, -0.5, -0.1, 0.4, the positive zero crossings would be 0.2 and -0.1.

20 Figure 24 shows a waveform with the positive zero crossings highlighted. They may not always lie on the zero line due to their sampling position. The samples which is closest to the zero line, in other words have the smallest absolute value, are always chosen. A full cycle, as shown for example in Figure 24, is made up of the samples lying between two consecutive positive zero crossings.

25

In the example shown the cycles are delimited with respect to the positive zero crossing. However, the cycles are not limited in this respect and may be delimited with respect to other criteria, such as negative zero crossings, peak values, etc. The only limitation is that preferably, both the test cycle and the
30 reference cycle are selected according to the same criteria.

With reference to Figure 22, the next step (step 1) is to choose a cycle beginning from the start of the data, to be the original cycle, x_0 . The values of the data of the samples in the original cycle, x_0 , are stored in the original cycle register 251.

- 5 A mismatch count, cx , stored in a mismatch counter 253, and a count of the number of data comparisons, lx , stored in a comparison counter 252, are both set to zero (step 2).

- 10 The next step (step 3) is to randomly pick another cycle, y_0 , elsewhere in the waveform, within a certain domain (parameter: comparison domain), to be the comparing reference cycle. Usually, the original cycle and the reference cycle would come from data having the same origin. However, the invention is not limited in this respect. For the cases where the waveform has a form where the comparison domain may be large, for example, waveforms, for example derived
15 from a running engine, which do not vary dramatically over time, the algorithm may be used to compare a test cycle from data from one source with a reference cycle from a second source. For cases, where the comparison domain may not be too large, for example, musical data which varies greatly over a short period of time, comparing a test source with a second reference source of data may not be so
20 satisfactory. Reference is made to Result 10a shown in Figure 38.

- Returning to Figure 22, the test cycle and the comparison cycle are then compared (steps 4, 5, 6, 7, 8) in order to obtain a mismatch score for the reference cycle, y_0 , with respect to the original cycle, x_0 . As seen in Figures 24 and 25, each
25 cycle, x_0 , y_0 includes a plurality of data samples or elements each having a value, s_j , s_j' , respectively. Each value having also a respective magnitude.

- The comparison of the cycles includes a series of steps and involves determining various quantities derived from the data in the cycles. The calculation
30 processor 26 carries out a series of calculations. The derived quantities are stored in registers 257, 258 and 259. Firstly, an integration value is obtained for the original cycle and the reference cycle. This, may for example, be the area of the original

cycle, $\sigma|s_j|$, and the area of the reference cycle, $\sigma|s'_j|$ (step 4). With reference to Figure 25, the area of a cycle is defined by the sum of the magnitudes of the individual samples in the cycle. Due to the definition of the area, which is the sum of the samples in the cycle, the area of identical cycles may vary to a great extent if the sampling rate is low and the waveform frequency is large. Hence, while using the cycle comparison algorithm, it is preferable to use at least 11kHz sampling frequency for acceptable accuracy and sensitivity.

With reference to Figure 25, which shows an example, the next step (step 5) is to derive a quantity which gives an indication of the extent of the difference between the area and the shape of the reference cycle, y_0 , with respect to the original cycle, x_0 . This is defined by the sum of the magnitudes of the difference between each of the corresponding samples in the original cycle and the reference cycle, $\sigma(|s_j - s'_j|)$. Figure 4 shows three graphs. The first graph 40 shows the original cycle, x_0 , having samples, s_j , having values s_1 to s_{14} . The area of the original cycle is equal to the sum of the magnitudes of the values, s_1 to s_{14} : that being $\sigma|s_j|$. The second graph 42 shows the reference cycle, y_0 , having samples, s'_j , having values s'_1 to s'_{14} . The area of the reference cycle is equal to the sum of the magnitudes of the values, s'_1 to s'_{14} : that being $\sigma|s'_j|$. The third graph 44 shows the difference the cycles as defined by $\sigma(|s_j - s'_j|)$.

The next step (step 6) is to establish whether both cycles have the same number of samples, s_j , s'_j . If the number of samples in the cycles are not equal, the shorter cycle is padded with samples of value zero until both the original and reference cycles contain the same amount of samples.

Figure 5 shows an example of the padding described above with respect to step 6 shown in Figure 1. In Figure 26, cycle 1 has nine samples while cycle 2 only has 6 samples. In order to do a comparison, both cycles are made equal in sample size. This is achieved by padding the cycle having the fewer number of samples. In the example shown in Figure 26, cycle 2 is padded with additional samples of value zero until it becomes the same size as the larger cycle, cycle 1 in this case.

The quantities derived in the steps described above are used to determine for each comparison of an original cycle with a reference cycle a "measure of difference" (step 8), which is a quantity that shows how different one cycle is from the other.

5 This empirical 'measure of difference' is defined as:

$$\text{Measure of Difference} = \frac{\text{Area Difference}}{\text{LargerAreaOfTwoCycles} + |\text{MaxArea} - \text{MinArea}|}$$

MaxArea is the largest area of a cycle in the entire comparison domain and *MinArea* is the smallest area of a cycle in the entire comparison domain.
 10 *LargerAreaOfTwoCycles* is the bigger area of the original cycle and the reference cycle.

The inventors have derived the definition of the "measure of difference" as shown above for the following reasons.

15 With reference to Figure 27, the first denominator, *LargerAreaOfTwoCycles*, is neutral to logarithmic increments of the cycle amplitude. This means that every time a cycle is compared against another geometrically similar cycle which is double its amplitude, the measure of difference is the same. For example when a sine cycle of amplitude 'X' is compared with another sine cycle of amplitude '2X', the
 20 measure of difference is 'D'. Hence when a sine cycle of amplitude 'X' is compared with another sine cycle of amplitude '½ X', the measure of difference would still be 'D'.

Further, with reference to Figure 28, the second denominator, $|\text{MaxArea} - \text{MinArea}|$, is a normalizing term for the quantity *AreaDifference* which is neutral to
 25 linear increments of the cycle amplitude. This means that if the amplitude of a geometrically similar cycle increases linearly, when a cycle is compared to the cycle next to itself, either left or right, both comparisons should give the same magnitude in the 'measure of difference'.

Either of these denominators may be chosen. It is not necessary to use both.
 30 However, if either of these denominations are used, it has been found that some desirable results as well as some undesirable ones occur. One of the denominators

tends to be more effective on certain waveforms than the other. Therefore, preferably, a hybrid denominator made by adding them together is chosen, as this results in a much more general and unbiased 'measure of difference' which is effective independent of the waveform.

5

The derived 'measure of difference' is next compared with a threshold value (step 9) to determine whether there is a mismatch.

If the calculated "measure of difference" for the original sample, x_0 , and the reference sample, y_0 , more than a certain threshold (PARAMETER: threshold), then
10 it is considered as being 'different'. The choice of the threshold can be varied, and will depend on the range of values within the set of data.

Further, with reference to Figures 22, when a mismatch occurs, the mismatch counter, cx , for the original sample, x_0 , is incremented (step 10).

15 When a match occurs the mismatch counter, cx , is not increased. The program returns to step 3 which creates a new random reference cycle, y_1 , before moving on to calculate the quantities described above in steps 4 and 5, and carrying out any necessary padding in step 6, before calculating the "measure of difference" in step 8.

20

For each original sample, x_0 , a certain number of comparisons, L , are made which result in a certain number of mismatches and matches. The total number of mismatches plus matches is equal to the number of comparisons (step 11 and step 14). The number of comparisons can be varied and will depend on the data to be
25 analysed and the processing power available. Also, the greater the number of comparisons, the greater the accuracy of the anomaly detection.

Each original cycle, x_0 , is compared with a certain number of reference samples, y_0 . The comparison steps from selecting a reference sample (step 3) to
30 calculating the "measure of difference" (step 8) is carried out over a certain number of times (parameter: comparisons) Once the "measure of difference" (step 8) has been calculated for the certain number of reference samples, y_L , and the

comparison done the certain number of times, L , the program returns to step 1 to select a different original sample, x_1 and the mismatch counter value, cx , and the number of comparisons, L , is output for original sample, x_0 (step 15).

5 Whether original sample, x_0 , is judged to be an anomaly will depend on the number of mismatches in comparison to the number of comparisons, L . The normalised anomaly scores for each original sample, x_0 , are obtained by dividing the mismatch counter, cx , for each sample, x_0 , by the number of comparisons, L , which is also equal to the maximum mismatch count, so that the anomaly score
10 ranges from zero to one, with zero being 0% mismatch and one being maximum mismatch.

Figures 24 to 39 show results obtained using the cycle comparison algorithm. With reference to our copending unpublished patent applications IPD ref
15 A30114, A30174 and A30175, it is noted that the cycle comparison algorithm does not require parameter radius and parameter neighbourhood size.

The results show good anomaly discrimination with no saturation.

20 If the comparison domain is unspecified, it is assumed to be the entire data length. The results show in the lower part of the diagram the input data for analysis. The upper portion of the diagram shows the mismatch scores achieved for each sample using the cycle analysis algorithm described above with reference to Figures 22 to 28. In the upper portion, an anomaly is identified as being those
25 portions having the highest mismatch scores.

The results shown are for audio signals. However, the present invention may also be applied to any ordered set of data elements. The values of the data may be single values or may be multi-element values.

30

Result 1a shown in Figure 29 shows a data stream of 500 elements having a binary sequence of zeros and ones. The anomaly to be detected is a one bit error at

both ends of the data. In this example, the number of comparisons was 500, and the threshold was equal to 0.1. However, the choice of the threshold value in this case was not critical. The peaks in the upper portion of the graph show a perfect discrimination of the one bit errors at either end of the data sequence.

5

Result 2a shown in Figure 30 shows data stream having the form of a sine wave with a change in amplitude. In this example, the number of comparisons was 250 and the threshold was equal to 0.01. However, the choice of the threshold value in this case was not critical. The peaks in the upper portion of the graph show a perfect discrimination of the anomaly. The highest mismatch scores being for those portions of the data stream where the rate of change of amplitude is the greatest.

10

Result 3a shown in Figure 31 shows a data stream having the form of a sine wave with background noise and burst and delay error. In this example, the number of comparisons was 250, and the threshold was equal to 0.15. The peaks in the upper portion of the graph show a perfect discrimination of the anomalous cycles.

15

Result 4a shown in Figure 32 shows a data stream having the form of a 440kHz sine wave that has been clipped. The data has been sampled at a rate of 22kHz. In this example, the number of comparisons was 250, and the threshold was equal to 0.15. The peaks show a perfect discrimination of the anomalous cycles.

20

Result 5a shown in Figure 33 shows a data stream having the form of a 440kHz sine wave including phase shifts. The data has been sampled at a rate of 44kHz. In this example, the number of comparisons was 250 and the threshold was equal to 0.15. The peaks show a perfect discrimination of the anomalies.

25

Result 6a shown in Figure 34 shows a data stream having the form of a 440kHz sine wave that has been clipped. The data has been sampled at a rate of 44kHz. In this example, the number of comparisons was 250, and the threshold

30

was equal to 0.15. The peaks show a near perfect discrimination of the anomalous cycles.

Result 7a shown in Figure 35 shows a data stream having the form of a
5 440kHz sine wave that has been clipped. The data has been sampled at a rate of
11kHz. In this example, the number of comparisons was 250 and the threshold was
equal to 0.05. In this example, the threshold value is critical as due to the low
sampling rate. As discussed above, for signals that lie in the audio range, at a
frequency of around 440kHz, the sampling rate is preferably greater than 11kHz.
10 This is shown in the Result 6a. The results are less satisfactory due to the low
sampling rate. However, the algorithm would have performed much better at a
higher sampling rate.

Result 8a shown in Figure 36 shows a 440kHz waveform modulated at
15 220kHz with a sampling rate of 6kHz. In this example, the number of comparisons
was 500 and the threshold was 0.15. The results show that although the average
score has increased, score saturation has not occurred. The algorithm has still
identified the anomalous region.

20 Result 9a shown in Figure 37 shows data having a 440kHz amplitude
modulated sine wave. In this example, the sampling rate was 6kHz, the number of
comparisons was 250 and the threshold was 0.15. The results show good
discrimination of the anomalous cycles. It is noted that some striation effects are
evident.

25

Result 10a shown in Figure 38 shows real audio data comprising a guitar
chord with a burst of noise. In this example, the sampling rate was 11kHz, the
number of comparisons was 250 and the threshold was 0.015. Unlike the previous
results, the comparison domain was not the entire data length but was 175 cycles.
30 This was critical due to the morphing of cycles in this complex waveform. The
results show that the noise has been very well identified. It is further notices that

the attack and decay region, where the chord is struck and when it dies away, also score high attention (mismatch) scores, as would be expected.

The above examples show very good results. For many types of waveform
5 the cycle comparison algorithm described here is favoured over the sample analysis algorithm described with reference to Figures 1 to 21. However, it is to be noted that there are some waveforms that may be more suitable for analysis by the sample analysis algorithm, for example where in a waveform it is not considered anomalous for a small amplitude cycle to be adjacent a large amplitude cycle.

10

It has been noticed that the cycle comparison algorithm has problems identifying a misplaced cycle in a set of ordered cycles. This is because as long as the cycle is common in other parts of the waveform, it will not be considered as an anomaly regardless of its position. Thus, preferably, it is advantageous to take more
15 than one cycle into account while doing the comparison. Thus, the original cycle, x_0 , may be a plurality of cycles. n subsequent cycles, x_n , together to do the comparison or to implement a random neighbourhood of cycles for comparison in the same way the algorithms described with reference to Figures 1 to 21 take a random neighbourhood of samples.

20

An error correction system is now described with reference to Figures 40 and 20, which has application to the present invention. Having used the anomaly detection system previously described to identify regions of anomaly in a waveform, error correction is provided to remove the detected errors. From the attention map
25 produced as described above, a suitable filter coefficient is set (PARAMETER: filter coefficient) so that only the anomalous region remains in the map before passing the data to an error correction algorithm. The data in the attention map is stored in registers.

30

The error correction algorithm used depends on the algorithm used to detect the anomaly. For the cycle comparison algorithm described above is for use together with a cutting and replacing correction algorithm. However, the sample

analysis algorithm described above with reference to Figures 1 to 21, it has been found that a shape learning error correction algorithm yields better results.

5 The cutting and replacement correction algorithm described below may be implemented directly. The success of the error correction however, is dependent primarily on being able to pinpoint the anomaly with confidence, which is the function of the detection algorithm.

10 Figure 39 shows the steps taken to perform the cutting cycles routine. This method cuts the erroneous regions away and joins the ends together. This reduces the chances of second order noise.

15 Figure 40 shows the steps taken to perform the replacing cycles routing. After the erroneous cycle is identified, the algorithm searches a certain number of cycles (parameter: search radius for replacement cycle) around the erroneous cycle for a cycle with the lowest score available. It then uses this cycle to replace the erroneous cycle. As with cutting cycles method, this method is best implemented if the cycle comparison algorithm is used for the detection.

20 A detection algorithm of the present invention has been demonstrated to be very tolerant to the type of input data as well as being very flexible in spotting anomalies in one-dimensional data. Therefore there are many applications where such detection method may be useful.

25 In the audio field, such a detection algorithm may be used as a line monitor to monitor recordings and playback for unwanted noise as well as being able to remove it. It may also be useful in the medical field as an automatic monitor for signals from a cardiogram or encephalogram of a patient. Apart from monitoring human signals, it may also be used to monitor engine noise. Like monitoring
30 humans, the output from machines, be it acoustic signals or electrical signals, deviate from its normal operating pattern as the machine's operating conditions vary, and in particular, as the machine approaches failure.

The algorithm may also be applied to seismological or other geological data and data related to the operation of telecommunications systems, such as a log of accesses or attempted accesses to a firewall.

As the detection algorithm is able to give a much earlier warning in the case of systems that are in the process of failing, in addition to monitoring and removing errors, it may also be used as a predictor. This aspect has application for example, in monitoring and predicting traffic patterns.

The invention can be described in generally terms as set out in the set of numbered clauses below:

1. A method of recognising anomalies contained within a set of data derived from an analogue waveform, the data represented by an ordered sequence of data elements each having a value, in respect of at least some of said data elements, including the steps of: selecting a group of test elements comprising at least two elements of the sequence; selecting a group of comparison elements comprising at least two elements of the sequence, wherein the comparison group has the same number of elements as the test group and wherein the elements of the comparison group have relative to one another the same positions in the sequence as have the elements of the test group; comparing the value of each element of the test group with the value of the correspondingly positioned element of the comparison group in accordance with a predetermined threshold to produce a decision that the test group matches or does not match the comparison group; selecting further said comparison groups and comparing them with the test group; generating a distinctiveness measure as a function of the number of comparisons for which the comparison indicates a mismatch.
2. A method according to clause 1 including the further step of : identifying ones of said positional relationships which give rise to a number of consecutive mismatches which exceeds said threshold.
3. A method according to clause 2 including the further steps of: storing a definition of each such identified relationship; and utilising the stored definitions for the processing of further data.
4. A method according to clause 2 or clause 3 including the further step of: replacing said identified ones

with data which falls within the threshold. 5. A method according to any preceding clause, wherein the time resolved data is an audio signal. 6. A method of removing noise from a sequence of data represented by an ordered sequence of data elements each having a value comprising, in respect of at least some of said data elements, including the steps of: selecting a group of comparison elements comprising at least two elements of the sequence, wherein the comparison group has the same number of elements as the test group and wherein the elements of the comparison group have relative to one another the same positions in the sequence as have the elements of the test group; comparing the value of each element of the test group with the value of the correspondingly positioned element of the comparison group in accordance with a predetermined match criterion to produce a decision that the test group matches or does not match the comparison group; selecting further said comparison groups and comparing them with the test group; generating a distinctiveness measure as a function of the number of comparisons for which the comparison indicates a mismatch, identifying ones of said positional relationships which give rise to a number of consecutive mismatches which exceeds a threshold, and replacing said identified ones with data which falls within the threshold. 7. A computer programmed to perform the method of any of clauses 1-6. 8. A computer program product directly loadable into the internal memory of a digital computer, comprising software code portions for performing the steps of any of clauses 1-6, when said product is run on a computer. 9. An apparatus for recognising anomalies contained within a set of data derived from an analogue waveform, the data represented by an ordered sequence of data elements each having a value comprising, in respect of at least some of said data elements, including: means for storing an ordered sequence of data, each datum having a value, means for selecting a group of test elements comprising at least two elements of the sequence; means for selecting a group of comparison elements comprising at least two elements of the sequence, wherein the comparison group has the same number of elements as the test group and wherein the elements of the comparison group have relative to one another the same positions in the sequence as have the elements of the test group; means for comparing the

value of each element of the test group with the value of the correspondingly positioned element of the comparison group in accordance with a predetermined match criterion to produce a decision that the test group matches or does not match the comparison group; means for selecting further said comparison groups and comparing them with the test group; means for generating a distinctiveness measure as a function of the number of comparisons for which the comparison indicates a mismatch.

10. A computer program product stored on a computer usable medium, comprising: computer readable program means for causing a computer to store an ordered sequence of data derived from an analogue waveform, each datum having a value, computer readable program means for causing a computer to select a group of test elements comprising at least two elements of the sequence; computer readable program means for causing a computer to select a group of comparison elements comprising at least two elements of the sequence, wherein the comparison group has the same number of elements as the test group and wherein the elements of the comparison group have relative to one another the same positions in the sequence as have the elements of the test group; computer readable program means for causing a computer to compare the value of each element of the test group with the value of the correspondingly positioned element of the comparison group in accordance with a predetermined match criterion to produce a decision that the test group matches or does not match the comparison group; computer readable program means for causing a computer to select further said comparison groups and comparing them with the test group; computer readable program means for causing a computer to generate a distinctiveness measure as a function of the number of comparisons for which the comparison indicates a mismatch.

11. A method of recognising anomalies in data represented by an ordered array of data elements each having a value, in respect of at least some of said data elements, including the steps of: selecting a group of test elements comprising at least two elements of the array; selecting a group of comparison elements comprising at least two elements of the array, wherein the comparison group has the same number of elements as the test group and wherein the elements of the comparison group have relative to one another the same positions in the array as have the elements of the test group;

comparing the value of each element of the test group with the value of the correspondingly positioned element of the comparison group in accordance with a dynamic threshold, whose value varies in accordance with the values of the elements around at least one of said test elements,

5 to produce a decision that the test group matches or does not match the comparison group; selecting further said comparison groups and comparing them with the test group; generating a distinctiveness measure as a function of the number of comparisons for which the comparison indicates a mismatch. 12. A method according to clause 1, including the further step of:

10 determining the local gradient at one of said test elements. 13. A method according to clause 2, including the further step of: using said local gradient to determine the dynamic threshold. 14. A method according to any of the preceding clauses wherein the dynamic threshold is determined in accordance with the local gradient and a predetermined threshold.

15 15. A method according to clause 1, including the further step of: determining the value of the elements neighbouring one of said test elements. 16. A method according to clause 6, wherein the dynamic threshold is determined in accordance with said value of the elements neighbouring one of said test elements. 17. A method according to clause 1 including the further step of :

20 identifying ones of said positional relationships which give rise to a number of consecutive mismatches which exceeds said threshold. 18. A method according to clause 7 including the further steps of: storing a definition of each such identified relationship; and utilising the stored definitions for the processing of further data. 19. A method according to clause 7 or clause 8 including the

25 further step of: replacing said identified ones with data which falls within the threshold. 20. A computer programmed to perform the method of any of clauses 11-19. 21. A computer program product directly loadable into the internal memory of a digital computer, comprising software code portions for performing the steps of any of clauses 11-19, when said product is run on a

30 computer. 22. An apparatus for recognising anomalies in data represented by an ordered array of data elements each having a value, in respect of at least some of said data elements, including: means for storing an ordered array of data,

each datum having a value, means for selecting a group of test elements comprising at least two elements of the array; means for selecting a group of comparison elements comprising at least two elements of the array, wherein the comparison group has the same number of elements as the test group and wherein the elements of the comparison group have relative to one another the same positions in the array as have the elements of the test group; means for comparing the value of each element of the test group with the value of the correspondingly positioned element of the comparison group in accordance with a dynamic threshold to produce a decision that the test group matches or does not match the comparison group; means for selecting further said comparison groups and comparing them with the test group; means for generating a distinctiveness measure as a function of the number of comparisons for which the comparison indicates a mismatch.

23. An apparatus according to clause 22, including means for determining the local gradient at one of said test elements.

24. An apparatus according to clause 23, including means for determining the dynamic threshold using said local gradient.

25. An apparatus according to any of clauses 22-24, wherein dynamic threshold is determined in accordance with the local gradient and a predetermined threshold.

26. An apparatus according to clause 22 including means for determining the value of the elements neighbouring one of said test elements.

27. An apparatus according to clause 26, wherein the dynamic threshold is determined in accordance with said value of the elements neighbouring one of said test elements.

28. An apparatus according to clause 22 including means for identifying ones of said positional relationships which give rise to a number of consecutive mismatches which exceeds said threshold.

29. An apparatus according to clause 28 including means for storing a definition of each such identified relationship; and utilising the stored definitions for the processing of further data.

30. An apparatus according to clause 28 or 29 including means for replacing said identified ones with data which falls within the threshold.

31. A computer program product stored on a computer usable medium, comprising: computer readable program means for causing a computer to store an ordered array of data, each datum having a value, computer readable program means for causing a

computer to select a group of test elements comprising at least two elements of the array; computer readable program means for causing a computer to select a group of comparison elements comprising at least two elements of the array, wherein the comparison group has the same number of elements as the test group and wherein the elements of the comparison group have relative to one another the same positions in the array as have the elements of the test group; computer readable program means for causing a computer to compare the value of each element of the test group with the value of the correspondingly positioned element of the comparison group in accordance with a dynamic threshold to produce a decision that the test group matches or does not match the comparison group; computer readable program means for causing a computer to select further said comparison groups and comparing them with the test group; computer readable program means for causing a computer to generate a distinctiveness measure as a function of the number of comparisons for which the comparison indicates a mismatch.

32. A method of recognising anomalies in data represented by an ordered array of data elements each having a value, in respect of at least some of said data elements, including the steps of: i) selecting a first test element from said array, ii) selecting a random reference element from said array, iii) comparing the value of the test element with the value of the random reference element, iv) if the value of said test element does not match the value of said random reference element searching for a matching element within the neighbourhood of said random reference element, v) changing a mismatch parameter as a measure of anomalies in said data array if no matching element within said neighbourhood of said random reference element is found and selecting a new random reference element, vi) repeating steps iii) to v) a number of times.

33. A method according to clause 32 including the steps of : vii) if in step iv) a matching element is found within said neighbourhood of said random reference element performing a comparison of the values of elements of a group of elements about said first test element with the values of a corresponding group of elements about said matching element, viii) if said groups are found to match increasing a comparison value.

34. A method according to clause 33 wherein said elements of said group of elements about said first test element and said elements

of said group of elements about said matching element are arranged in the same manner about said test element and said matching element respectively and corresponding elements of said groups are compared in accordance with a threshold value.

35. A method according to clause 33 in which step vi) is repeated until said comparison value is equal to a set value and when said comparison value is equal to said set value selecting a second test element and repeating steps i) to vi) for said second test element.

36. A method according to clause 34, wherein the values are compared in accordance with a dynamic threshold, the value of which varies in accordance with the values of the elements around at least one of the test elements.

37. A method according to clause 36, including the further step of:
determining the local gradient at one of said test elements.

38. A method according to clause 39, including the further step of:
using said local gradient to determine the dynamic threshold.

39. A method according to any of preceding clauses 36 to 38 wherein the dynamic threshold is determined in accordance with the local gradient and a predetermined threshold.

40. A method according to clause 34 including the further step of : identifying the particular arrangements of elements which give rise to a number of consecutive mismatches which exceeds said threshold and storing data representing such particular arrangements of elements.

41. A method according to clause 40 including the further step of: replacing said stored data with corresponding data of arrangements giving rise to matches falling within the threshold.

42. A computer programmed to perform the method of any of clauses 31-41.

43. A computer program product directly loadable into the internal memory of a digital computer, comprising software code portions for performing the steps of any of clauses 31-41, when said product is run on a computer.

44. An apparatus for recognising anomalies in data represented by an ordered array of data elements each having a value, in respect of at least some of said data elements, means for selecting a first test element from said array , means for selecting a random reference element from said array, means for comparing the value of the test element with the value of the random reference element, means for searching for a matching element within the neighbourhood of said random

reference element if the value of said test element does not match the value of said random reference element, means for changing a mismatch parameter as a measure of anomalies in said data array if no matching element is found within said neighbourhood of said random reference element and for selecting a new random reference element.

5 45. An apparatus according to clause 44, wherein if a matching element is found within said neighbourhood of said random reference element means are provided to perform a comparison of the values of elements of a group of elements about said first test element with the values of a corresponding group of elements about said matching element, and if said groups are found to match means are provided to increase a comparison value.

10 46. An apparatus according to clause 45 wherein said elements of said group of elements about said first test element and said elements of said group of elements about said matching element are arranged in the same manner about said test element and said matching element respectively and corresponding elements of said groups are compared in accordance with a threshold value.

15 47. An apparatus according to clause 45, including means for repeating step vi) until said comparison value is equal to a set value and when said comparison value is equal to said set value selecting a second test element and including means for repeating steps i) to vi) for said second test element.

20 48. An apparatus according to clause 46, wherein the values are compared in accordance with a dynamic threshold, the value of which varies in accordance with the values of the elements around at least one of the test elements.

49. An apparatus according to clause 48, including means for determining the local gradient at one of said test elements.

25 50. An apparatus according to clause 49 including means for using said local gradient to determine the dynamic threshold.

51. An apparatus according to any one of clauses 48-50, wherein the dynamic threshold is determined in accordance with the local gradient and a predetermined threshold.

52. An apparatus according to clause 46, including means for identifying the particular arrangements of elements which give rise to a number of consecutive mismatches which exceeds said threshold and storing data representing such particular arrangements of elements.

30 53. An apparatus according to clause 52, including means for replacing said stored data with corresponding data of arrangements giving rise to matches falling

within the threshold. 54. An apparatus according to clause 44 including means for identifying ones of said test elements which give rise to a number of consecutive mismatches which exceed said threshold. 55. An apparatus

according to clause 54 including means for storing a definition of each such test
5 elements; and utilising the stored test elements for the processing of further data.

56. An apparatus according to clause 54 or 55 including means for replacing said identified ones with data which falls within the threshold. 57. A computer

program product stored on a computer usable medium, comprising: computer
10 readable program means for causing a computer to store an ordered array of data elements each having a value, in respect of at least some of said data elements, computer readable program means for causing a computer to select a first test element from said array,

computer readable program means for causing a computer to select a random reference element from said array, computer readable program means for causing a computer to compare the value of the test
15 element with the value of the random reference element, computer readable program means for causing a computer to search for a matching element within the neighbourhood of said random reference element if the value of said test element does not match the value of said random reference element, computer readable program means for causing a computer to change a mismatch parameter as a

20 measure of anomalies in said data array if no matching element is found within said neighbourhood of said random reference element and for selecting a new random reference element. 58. A method of recognising anomalies contained within an array of data elements, each element having a value, including the steps of, in

respect of at least some of said data elements, i) identifying cycles in the set of
25 data in accordance with predetermined criteria, ii) selecting a test cycle of elements from said set of data, iii) randomly selecting a comparison cycle from said set of data, iv) determining an integration value for said test cycle and said reference cycle respectively, v) comparing said integration values and deriving therefrom a measure of the difference of said test and said reference cycles, vi)

30 using said measure to determine a mismatch of said test and said reference cycles. 59. A method according to clause 58, including the further step of: vii) randomly selecting further reference cycles and comparing them with the test

cycle according to steps v) and vi) and counting the number of mismatches.

60. A method according to clause 58 in which a mismatch is determined by comparing said measure to a threshold value.

61. A method according to clause 59, including the further step of: viii) generating a distinctiveness measure as a function of the number of mismatches between test and reference cycles.

62. A method according to any preceding clause, including the further step of: ix) establishing whether the test and reference cycles include the same number of elements, and if the number of elements are not equal, padding the cycle with fewer elements with elements of set value, so that both cycles contain the same number of elements.

63. A method according to any preceding clause, in which step iv) comprises determining the difference of the sums of values of the element of the test cycle and the comparison cycle respectively.

64. A method according to clause 59 in which step vii) is repeated a set number of times, after which a fresh test cycle is selected.

65. A computer programmed to perform the method of any of clauses 58 to 64.

66. A computer program product directly loadable into the internal memory of a digital computer, comprising software code portions for performing the steps of any of clauses 58 to 64, when said product is run on a computer.

67. An apparatus for recognising anomalies contained within an array of data elements, each element having a value, the apparatus including: means for identifying cycles in the set of data in accordance with predetermined criteria, means for selecting a test cycle of elements from said set of data, means for randomly selecting a comparison cycle from said set of data, means for determining an integration value for said test cycle and said reference cycle respectively, means for comparing said integration values and deriving therefrom a measure of the difference of said test and said reference cycles, means for using said measure to determine a mismatch of said test and said reference cycles.

68. An apparatus according to clause 67, further including: means for randomly selecting further reference cycles and comparing them with the test cycle, and means for counting the number of mismatches.

69. An apparatus according to clause 67, in which a mismatch is determined by comparing said measure to a threshold value.

70. An apparatus according to clause 68 or clause 69, further including: means for

generating a distinctiveness measure as a function of the number of mismatches between test and reference cycles. 71. An apparatus according to any of clauses 67 to 70, further including: means for establishing whether the test and reference cycles include the same number of elements, and if the number of elements are not equal, padding the cycle with fewer elements with elements of set value, so that both cycles contain the same number of elements. 72. An apparatus according to any of clauses 68 to 71, wherein said determining means determines the difference of the sums of values of the element of the test cycle and the comparison cycle respectively. 73. An apparatus according to clause 68, including means for selecting a fresh test cycle after the comparison means is repeated a predetermined number of times. 74. A computer program product stored on a computer usable medium, comprising: computer readable program means for causing a computer to identify cycles in the set of data in accordance with predetermined criteria, computer readable program means for causing a computer to select a test cycle of elements from said set of data, computer readable program means for causing a computer to randomly select a comparison cycle from said set of data, computer readable program means for causing a computer to determine an integration value for said test cycle and said reference cycle respectively, computer readable program means for causing a computer to compare said integration values and deriving therefrom a measure of the difference of said test and said reference cycles, computer readable program means for causing a computer to use said measure to determine a mismatch of said test and said reference cycles. 75. A computer program product stored on a computer usable medium according to clause 74, further comprising: computer readable program means for causing a computer to select further said comparison cycles and comparing them with the test cycle. 76. A computer program product stored on a computer usable medium according to either clause 74 or 75, further comprising: computer readable program means for causing a computer to generate a distinctiveness measure as a function of the number of comparisons for which the comparison indicates a mismatch.

Claims:

1. A method of recognising anomalies in data representative of an analogue waveform, the analogue waveform having a plurality of cycles, the data comprising an ordered sequence of data elements, each element having a respective value, the method including the steps of:
- (i) selecting a test group of test elements;
 - (ii) selecting a comparison group of comparison elements;
 - (iii) performing a comparison between the test group and the comparison group, the comparison involving the test elements of the test group on the one hand and the comparison elements of the comparison group on the other hand;
 - (iv) determining as a result of the comparison whether there is a match or a mismatch between the test group and the comparison group;
 - (v) repeating steps (ii), (iii), and (iv), incrementing the value of a mismatch counter each time a mismatch is found;
 - (vi) determining an anomaly measure representative of the anomaly of one or more of the test elements, the anomaly measure being dependent on value of the mismatch counter.
2. A method as claimed in claim 1, wherein a comparison value is generated as a result of the comparison between the test group and the comparison group, a mismatch being determined in dependence on the generated comparison value relative to a threshold value.

3. A method as claimed in claim 1 or claim 2, wherein the anomaly measure is the value of the mismatch counter.
4. A method as claimed in any of claims 1 to 3, wherein the steps (i) to (vi) are repeated so as to generate an anomaly measure for each of the elements in the sequence.
5. A method as claimed in any of the preceding claims, wherein steps (ii), (iii) and (iv) are repeated until a match is found between the test group and the comparison group.
6. A method as claimed in any of claims 1 to 4, wherein steps (ii), (iii) and (iv) are repeated a predetermined number of times.
7. A method as claimed in any preceding claim, wherein the test group includes a reference test element and the comparison group includes a reference comparison element, and wherein the comparison elements are selected such that the respective position of comparison elements in the sequence relative to the reference comparison element is the same as that of the test elements relative to the reference test element, the comparison involving comparing the value of each test element of the test group with the correspondingly positioned comparison element of the comparison group, the mismatch counter being incremented in dependence on the difference between the values of the correspondingly positioned elements in relation to a threshold value.
8. A method as claimed in claim 7, wherein the position in the sequence of the test elements relative to the reference test element is selected randomly from those elements within a predetermined neighbourhood range relative to the reference test element, and/or wherein the position of the reference comparison elements is selected randomly within a predetermined comparison range relative to the reference test element.

9. A method as claimed in claim 8, wherein if a match between the test group and a comparison group is found, the step of randomly selecting test elements within the predetermined neighbourhood range is repeated.

5 10. A method as claimed in any of claims 7 to 9, wherein the threshold value is dependent on the gradient of the waveform at the point in the waveform which the reference test element represents.

10 11. A method as claimed in any preceding claim, wherein the threshold value is dependent on the gradient of the waveform at some or each of the elements being used to perform a comparison between the elements of the test group and those of a comparison group.

15 12. A method as claimed in claim 7, wherein the difference in value of each pair of correspondingly positioned elements in the respective test group and comparison group are compared to a threshold value, the threshold value for each pair being dependent on the gradient of one or both elements of the pair.

20 13. A method as claimed in claim 11 or claim 12, wherein the gradient is equal to the difference in value of two adjacent elements.

25 14. A method as claimed in any preceding claim, including the further step of (a) determining if the value of the reference comparison element is within a predetermined range of the value of the reference test element, and if the value of the reference comparison is outside the predetermined range, (b) selecting again a reference comparison element.

30 15. A method as claimed in claim 14, wherein the steps (a), (b) of claim 14 are repeated until one of a plurality of stop conditions is met, the stop conditions including: (i) that a match is found between the test group and a comparison group; and (ii) that each element within a test range has been selected as a reference

comparison element, the mismatch counter being incremented when a stop condition is met.

16. A method as claimed in claim 14 or claim 15, wherein if a first comparison
5 reference element is selected that is outside the predetermined range, a second comparison reference element is selected that is a predetermined interval away in the ordered sequence from the first selected comparison reference element.

10 17. A method as claimed in claim 1, including the further step of identifying cycles in the set of data in accordance with predetermined criteria, wherein the test group of test elements is formed by one of the identified cycles, and the comparison group of comparison elements is formed by another of the identified cycles, and wherein the step of performing a comparison between the comparison group and
15 the test group includes determining a respective integration value for the test group and the comparison group, and comparing the integration values of each group.

18. A method as claimed in claim 17, wherein the step of performing a comparison between the comparison group and the test group involves determining
20 a respective combination of the values of the elements of the test group and those of the comparison group, and evaluating the difference in the respective combinations.

19. A method as claimed in claim 18, wherein the combination is a sum.
25

20. A computer programmed to perform the method of any of claims 1 to 19.

21. A computer program product directly loadable into the memory of a digital computer device, comprising software code portions for performing the steps of any
30 of claims 1 to 19, when the product is run on a computer device.

22. A computer program product stored on a computer-usable medium, the computer program product being configured for, in use, recognising anomalies in data representative of an analogue waveform, the analogue waveform having a plurality of cycles, the data comprising an ordered sequence of data elements, each
5 element having a respective value, the computer program product having:

computer-readable program means for selecting a test group of test elements;

10 computer-readable program means for selecting a comparison group of comparison elements;

computer-readable program means for performing a comparison between the test group and the comparison group, the comparison involving the test elements of the test group on the one hand and the comparison elements of
15 the comparison group on the other hand;

computer-readable program means for determining as a result of the comparison whether there is a match or a mismatches between the test group and the comparison group;

20 computer-readable program means for determining as a result of the comparison whether there is a match or a mismatches between the test group and the comparison group; and,

25 computer-readable program means for determining an anomaly measure representative of the anomaly of one or more of the test elements, the anomaly measure being dependent on value of the mismatch counter.

23. Apparatus for recognising anomalies in data representative of an analogue
30 waveform, the analogue waveform having a plurality of cycles, the data comprising an ordered sequence of data elements, each elements having a respective value, the apparatus including:

means for selecting a test group of test elements;

means for selecting a comparison group of comparison elements;

5

means for performing a comparison between the test group and the comparison group, the comparison involving the test elements of the test group on the one hand and the comparison elements of the comparison group on the other hand;

10

means for determining as a result of the comparison whether there is a match or a mismatches between the test group and the comparison group;

15

means for determining as a result of the comparison whether there is a match or a mismatches between the test group and the comparison group; and,

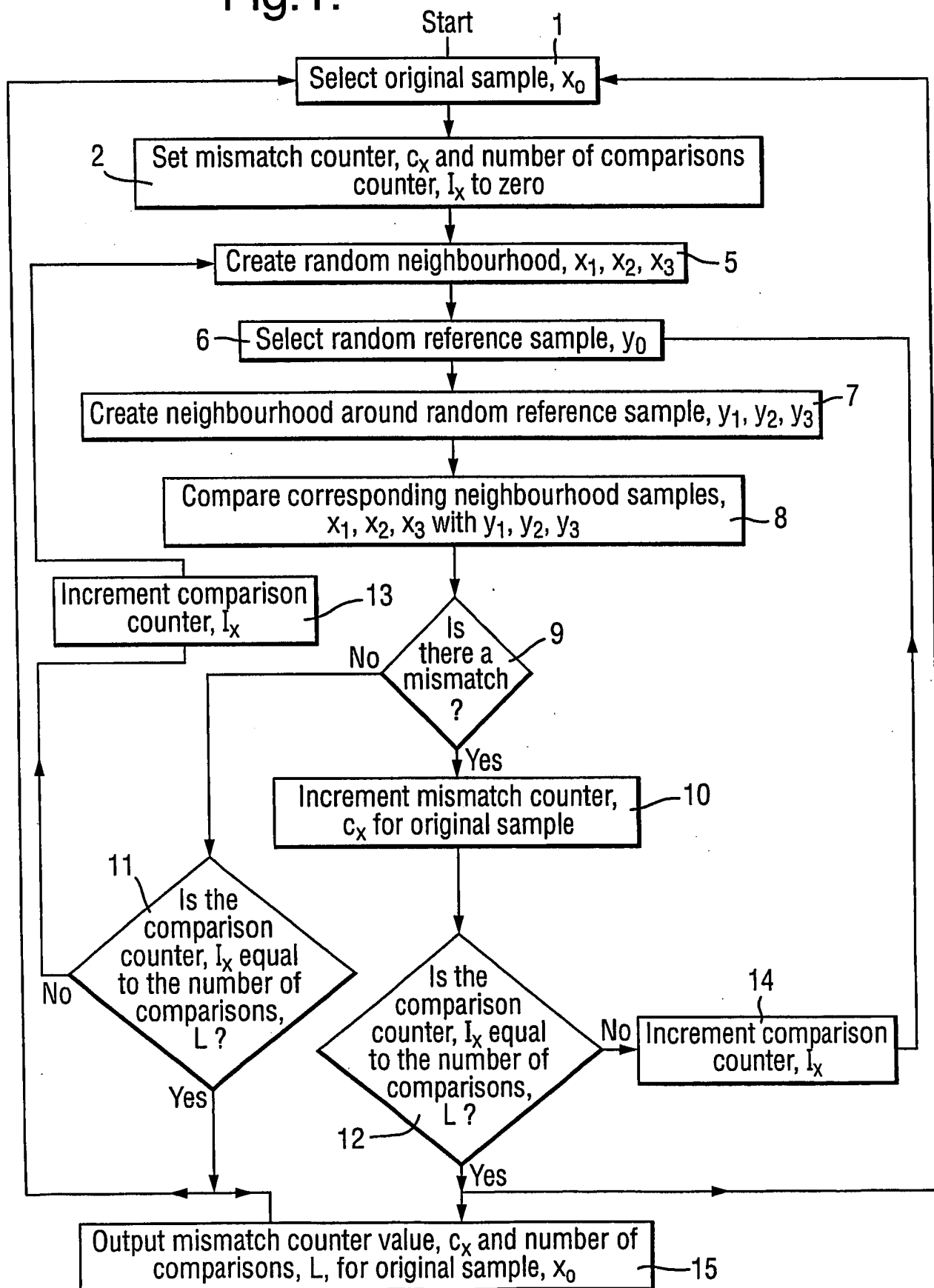
20

means for determining an anomaly measure representative of the anomaly of one or more of the test elements, the anomaly measure being dependent on value of the mismatch counter.

BEST AVAILABLE COPY

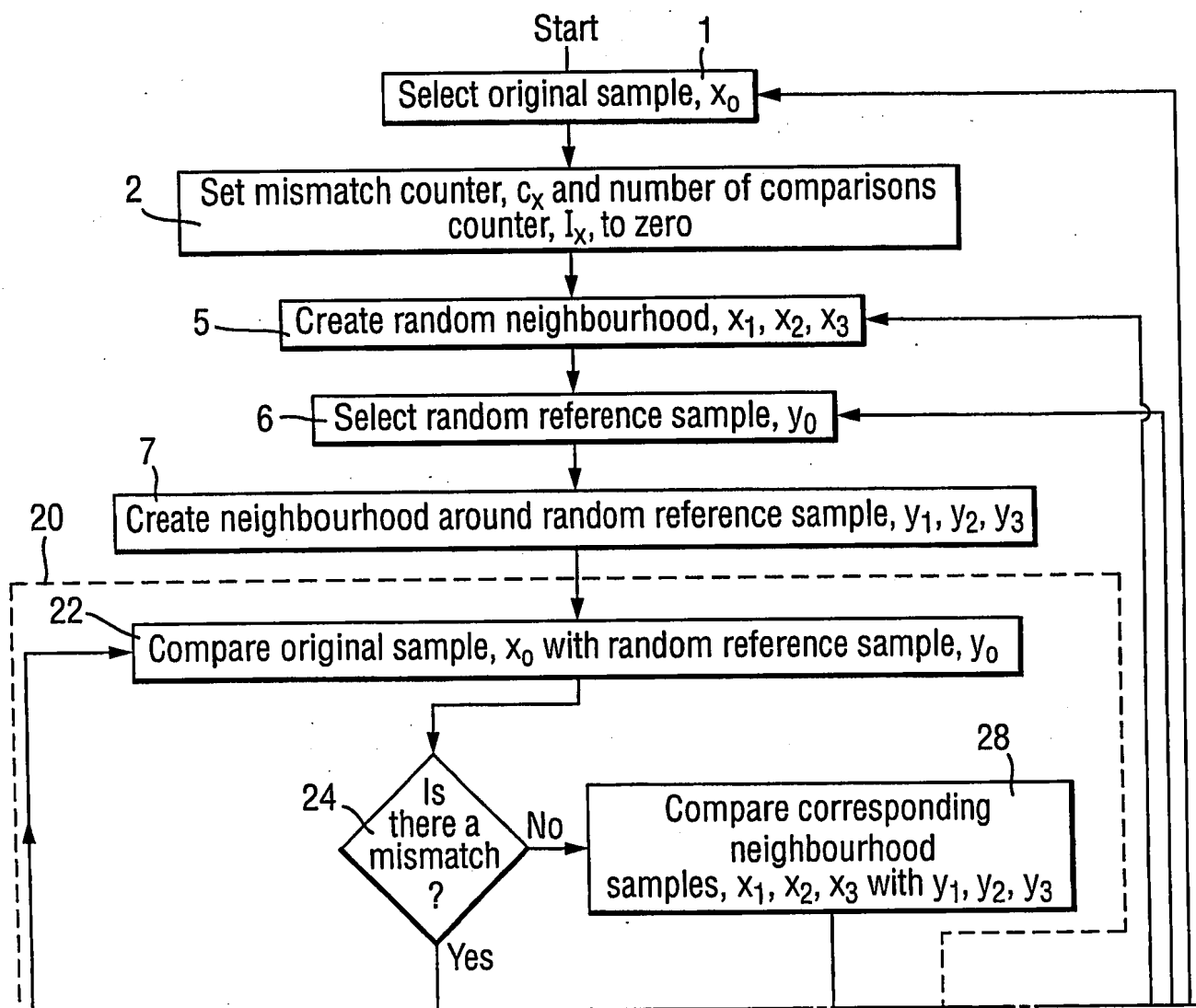
1/26

Fig.1.



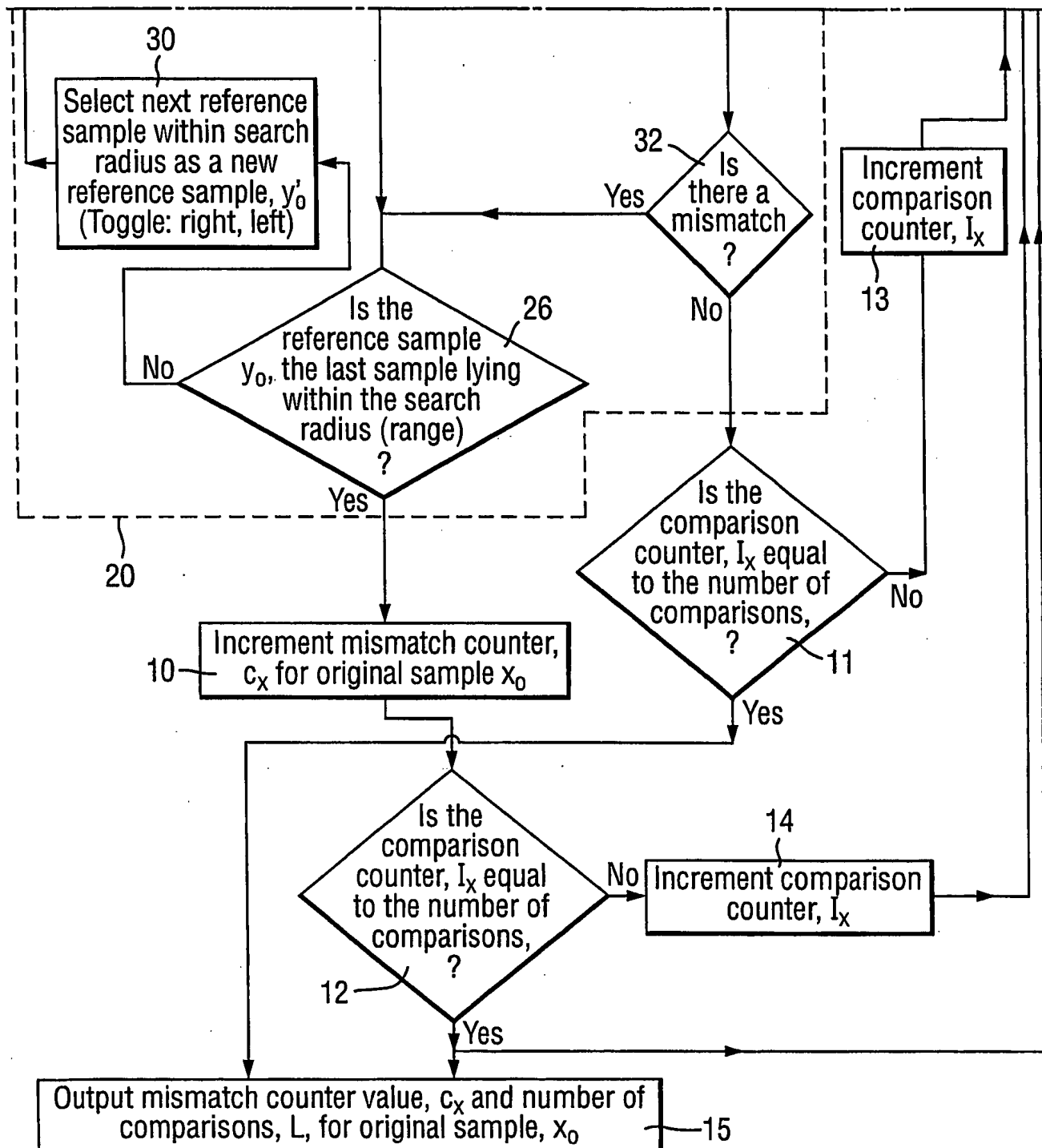
2/26

Fig.2.



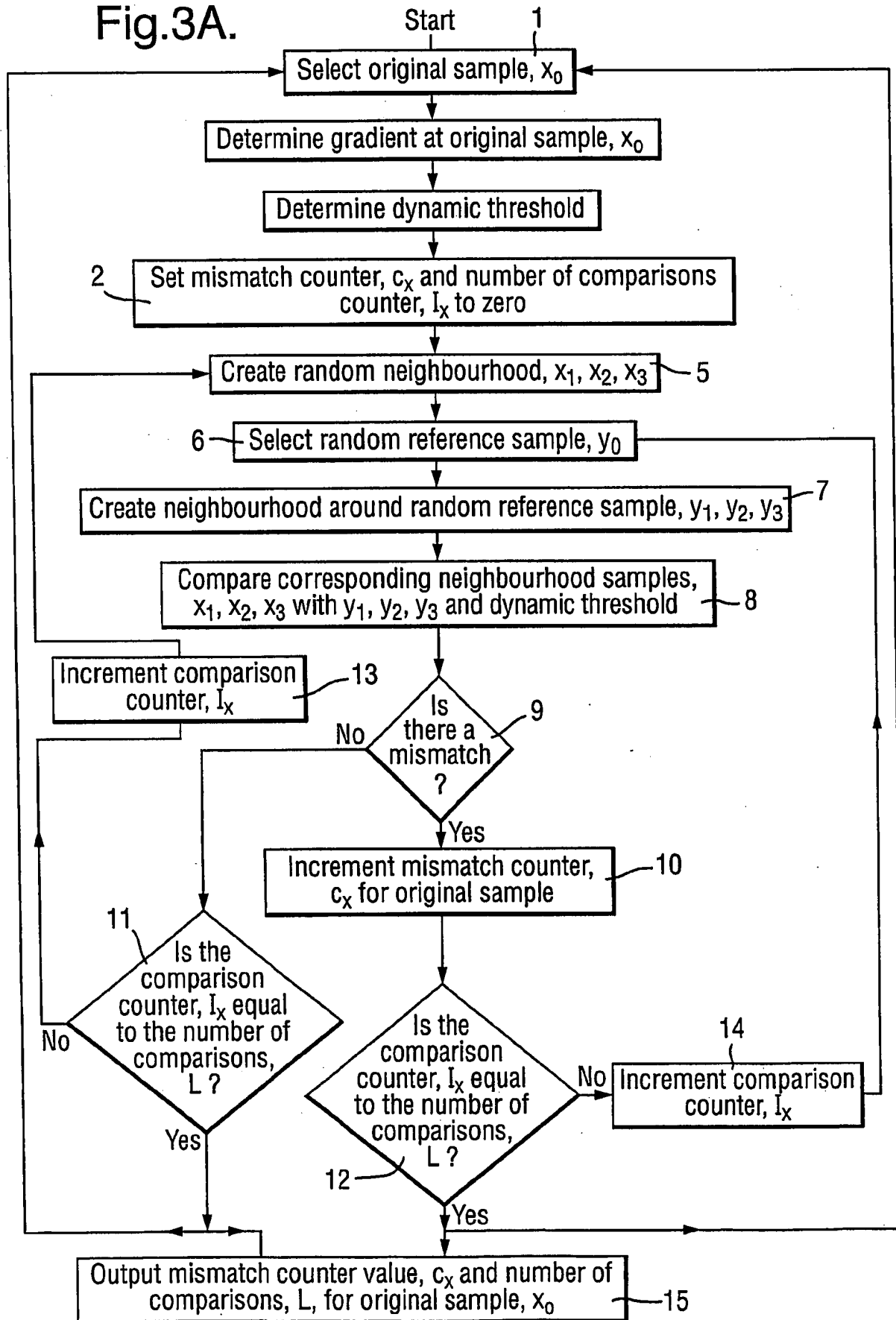
3/26

Fig.2(Cont.)



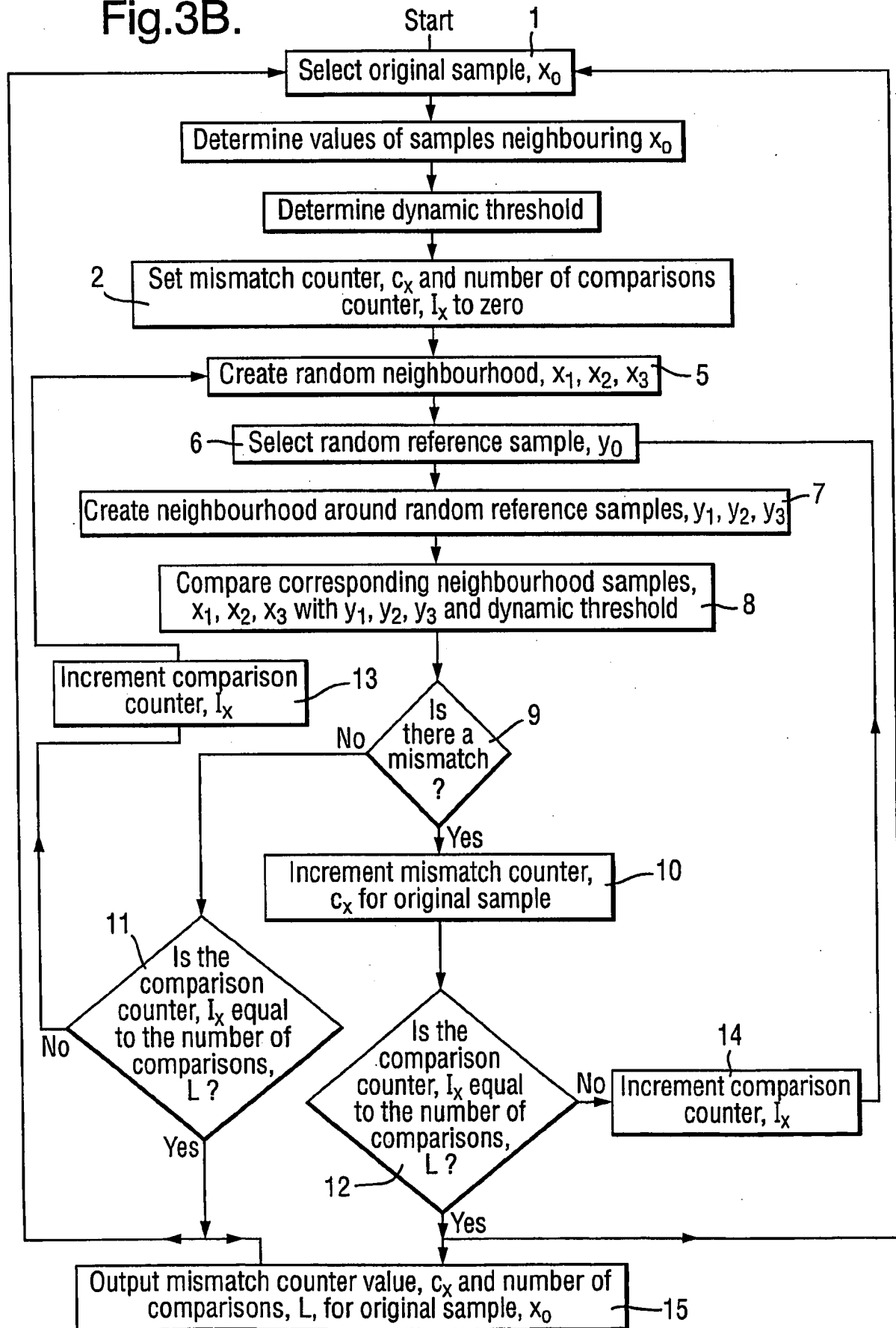
4/26

Fig.3A.



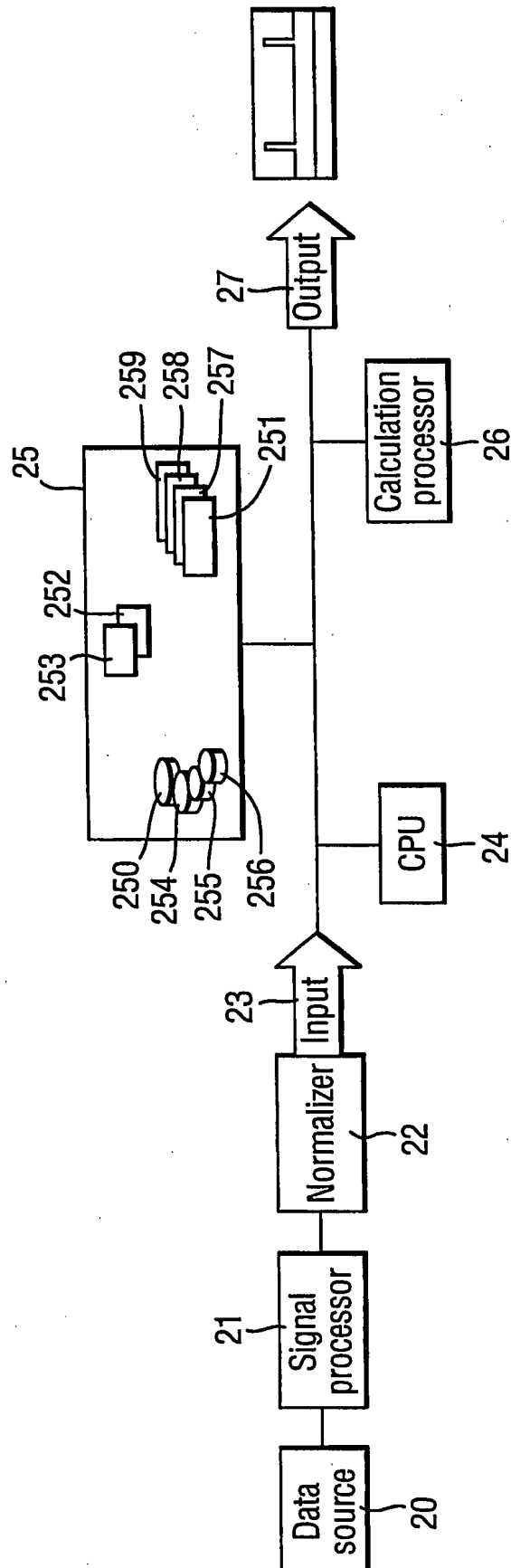
5/26

Fig.3B.



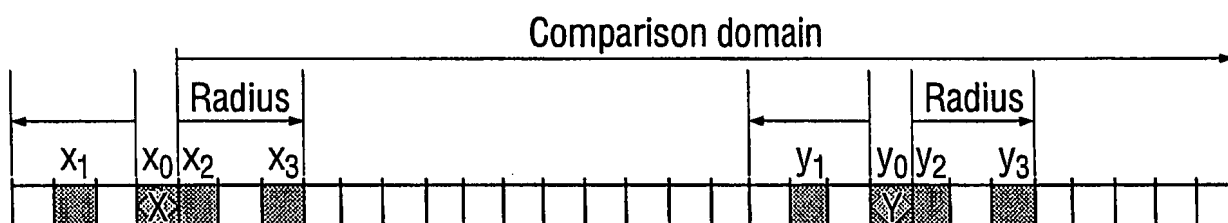
6/26

Fig.4.



7/26

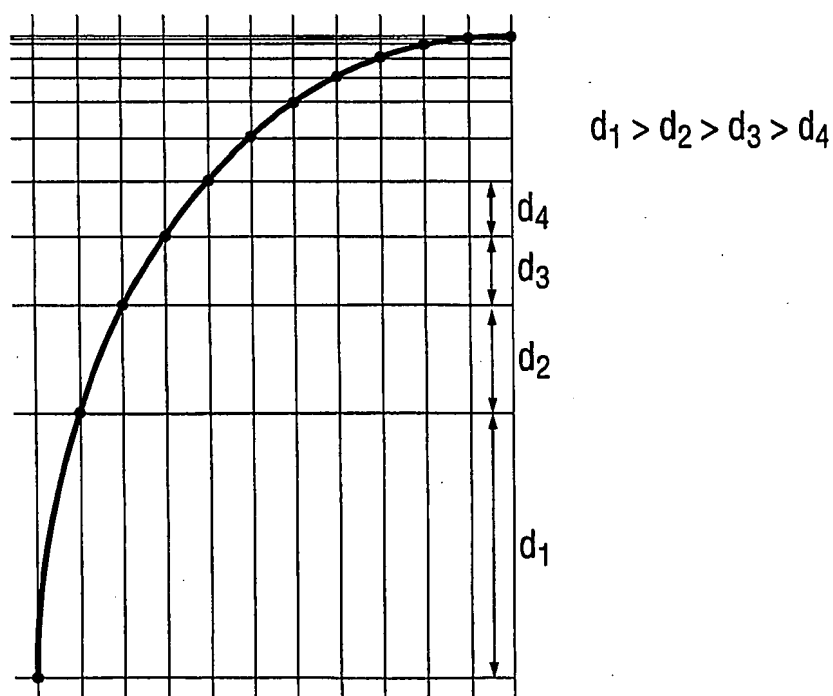
Fig.5.



$n : 0 \text{ to } 3$, Mismatch if $|x_n - y_n| > \text{threshold}$

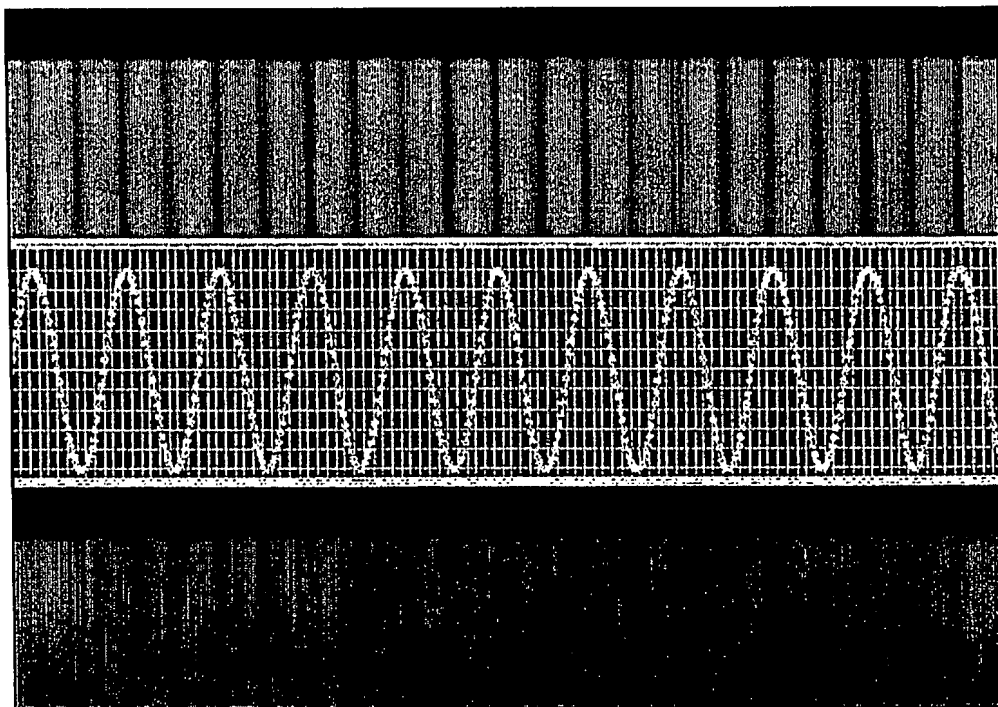
Radius = 3
 Neighbourhood size = 3
 Comparison domain = region where y is chosen

Fig.6.



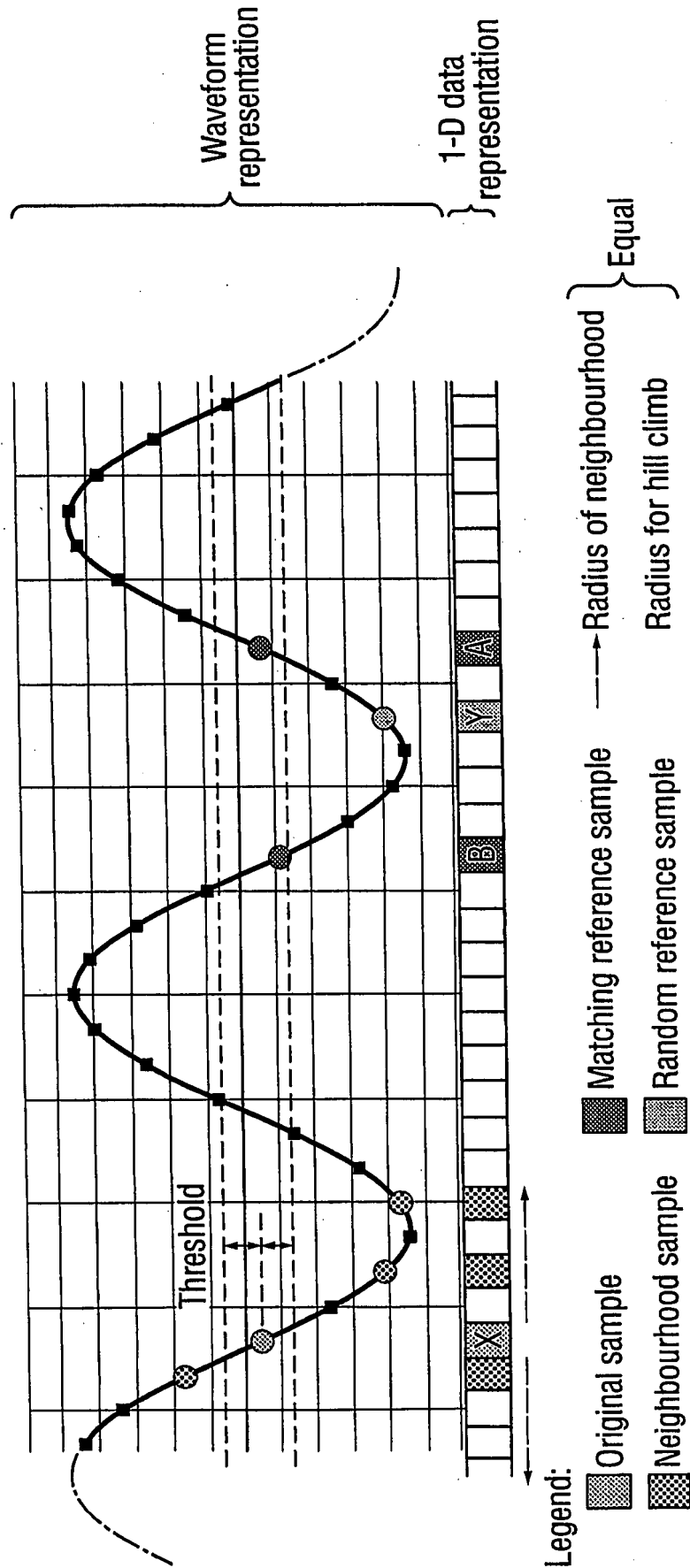
8/26

Fig.7.



9/26

Fig.8.



10/26

Fig.9.

Result 1

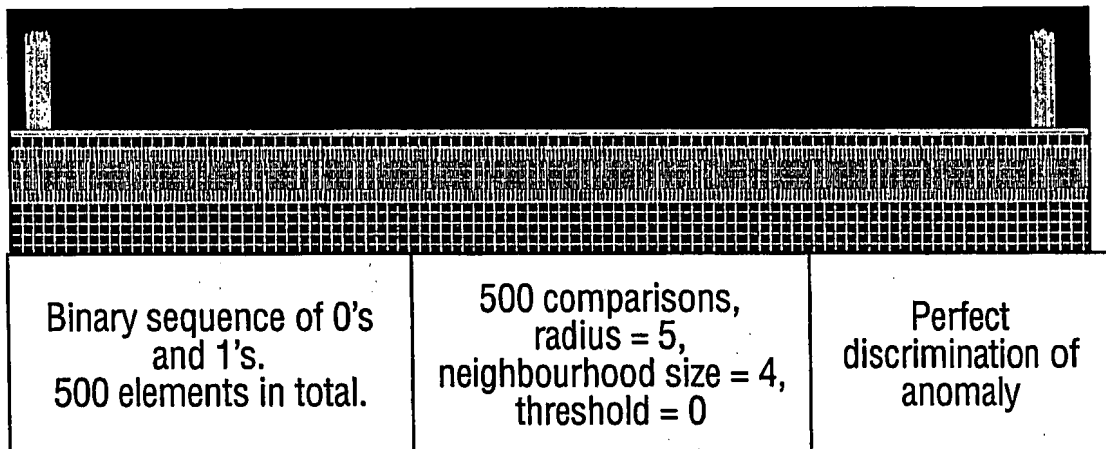


Fig.10.

Result 2

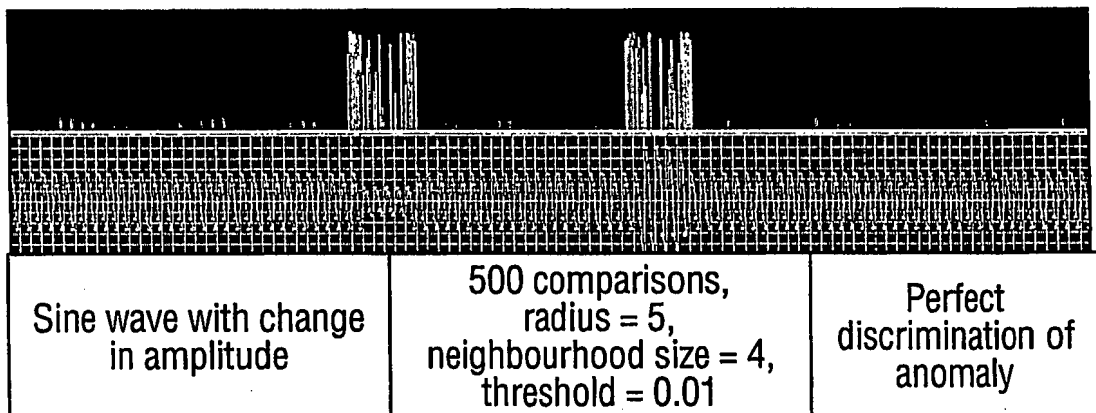
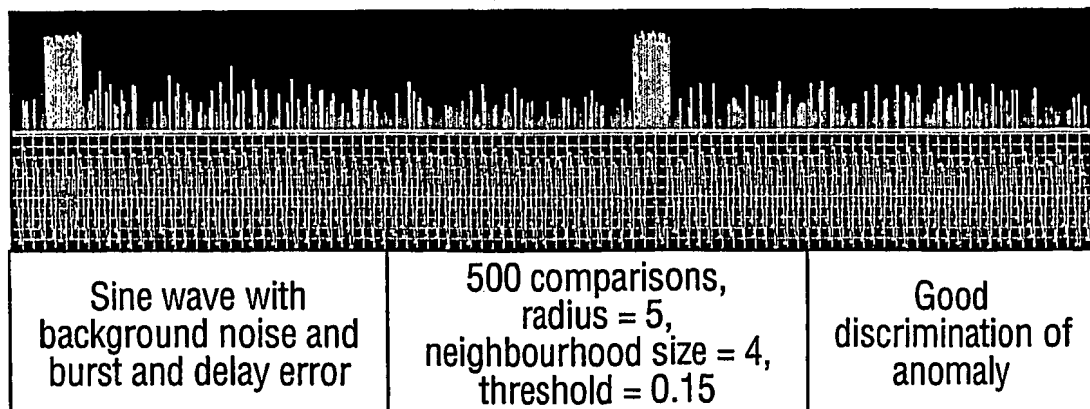


Fig.11.

Result 3



11/26

Fig.12.

Result 4

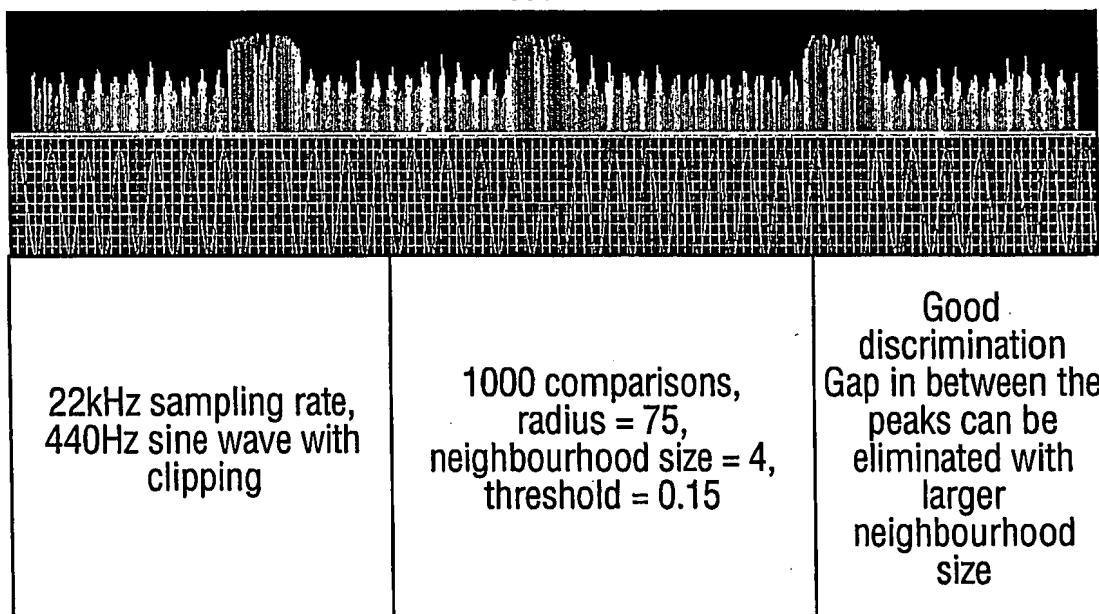
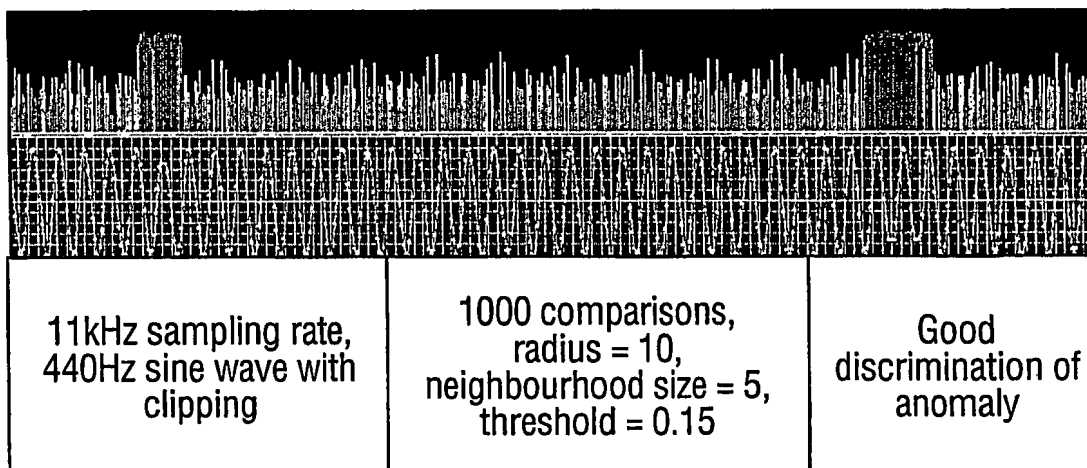


Fig.13.

Result 5



12/26

Fig.14.

Result 6

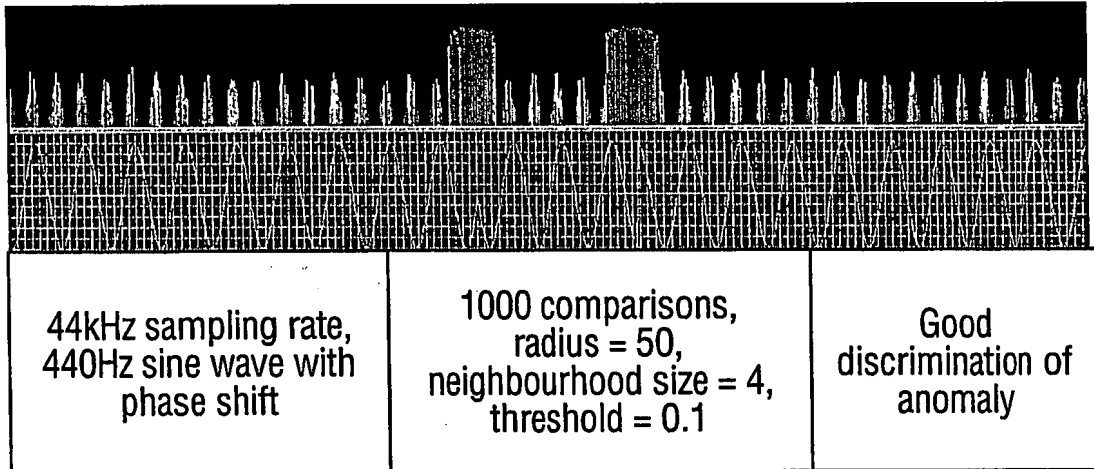


Fig.15.

Result 7

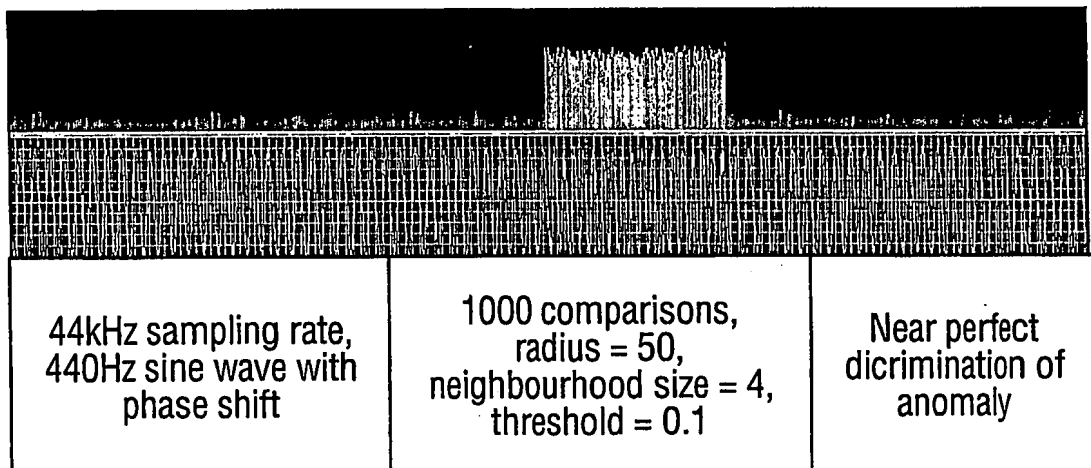


Fig.16.

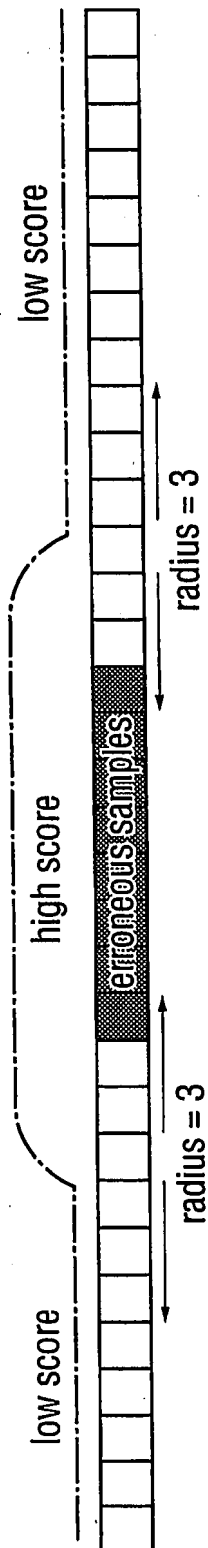


Fig.17.

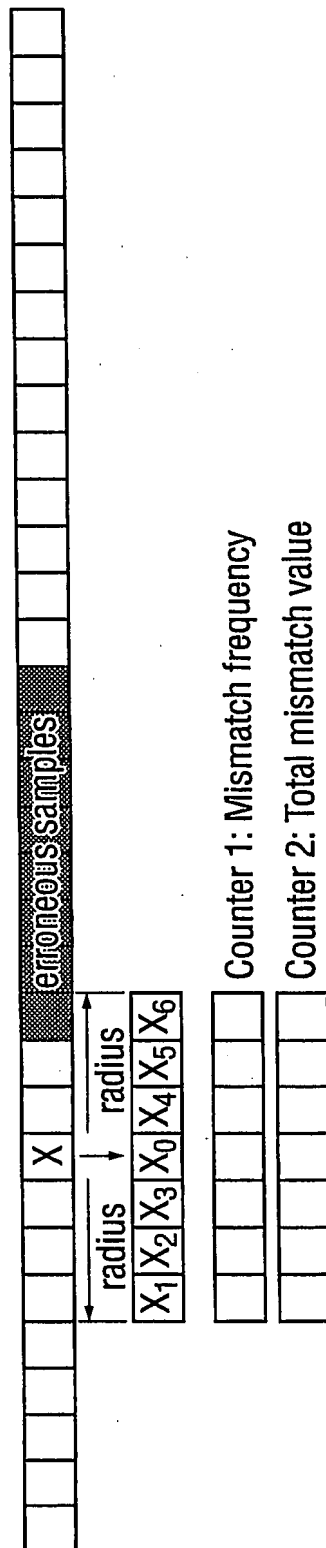


Fig.18.

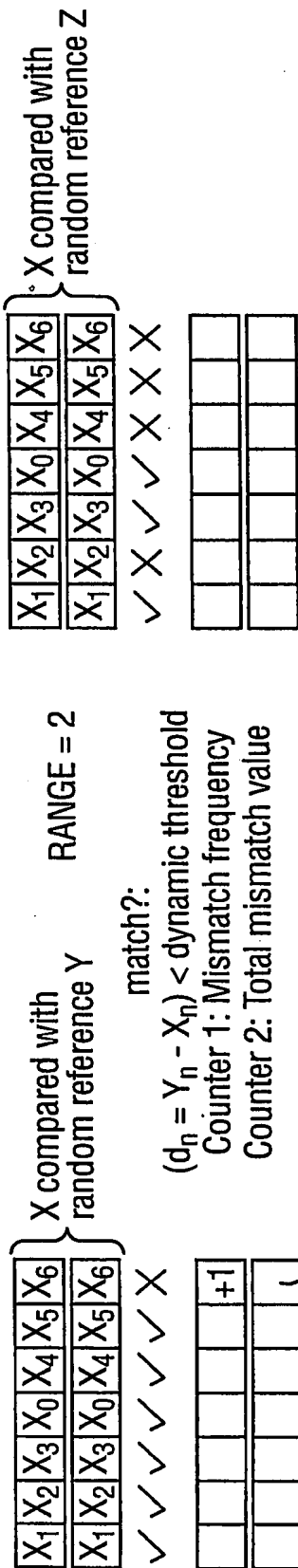
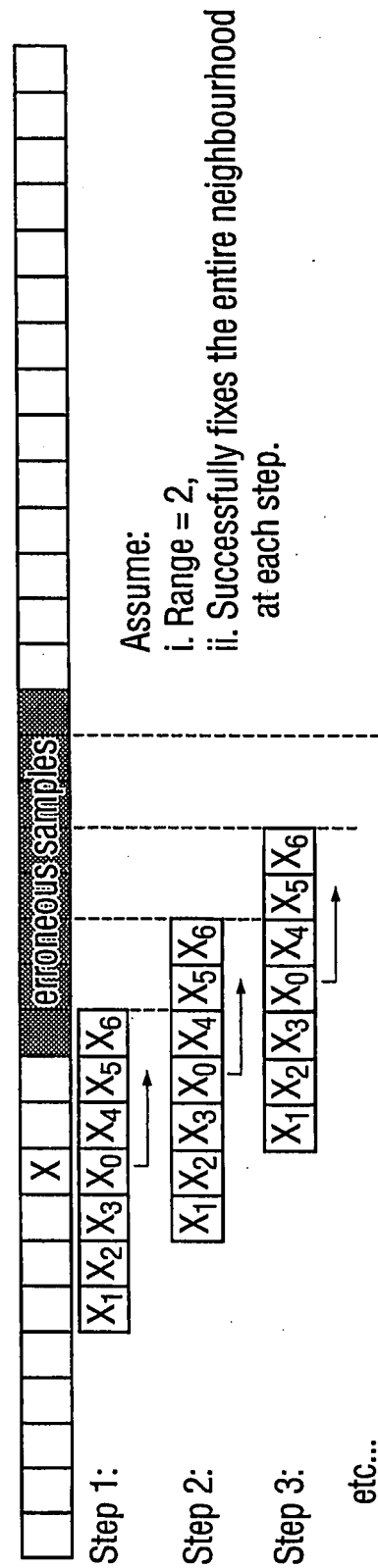
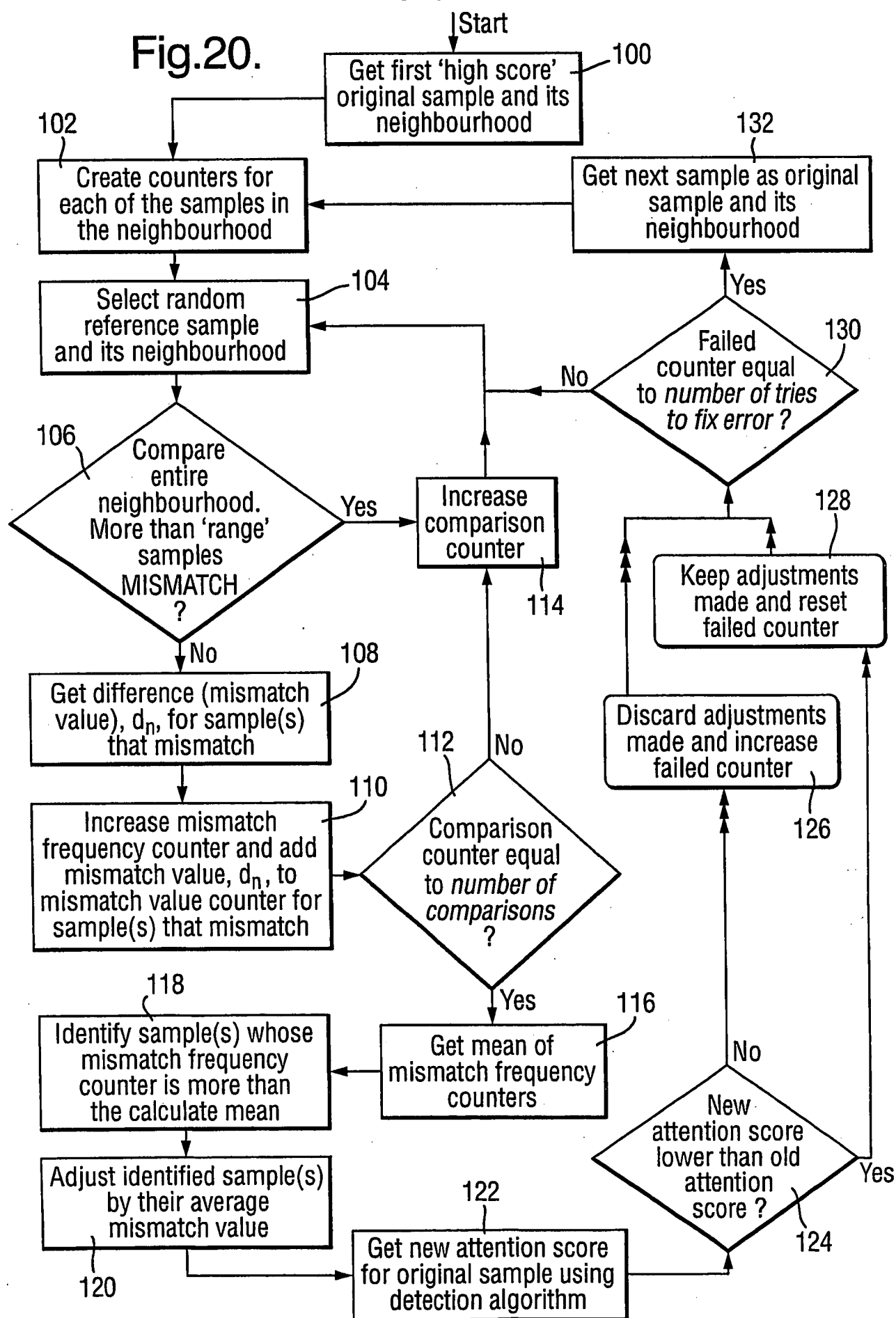


Fig.19.



15/26

Fig.20.



16/26

Fig.21.

Result 8

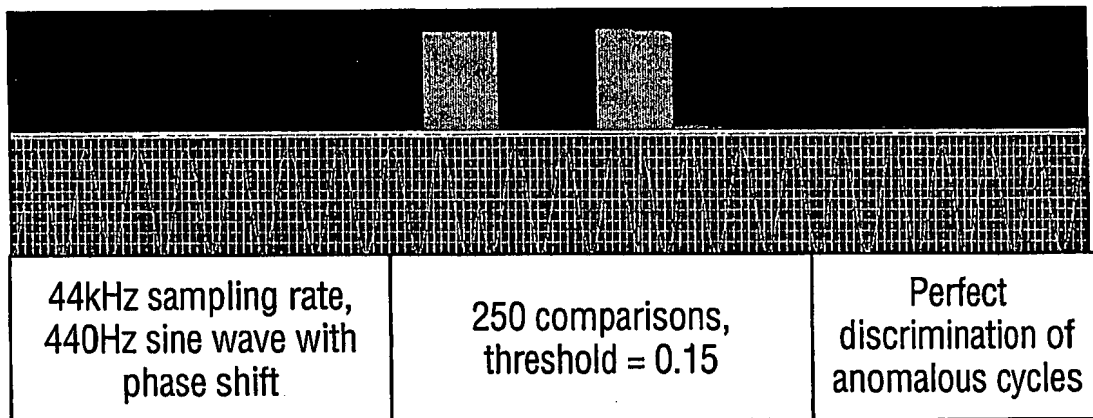
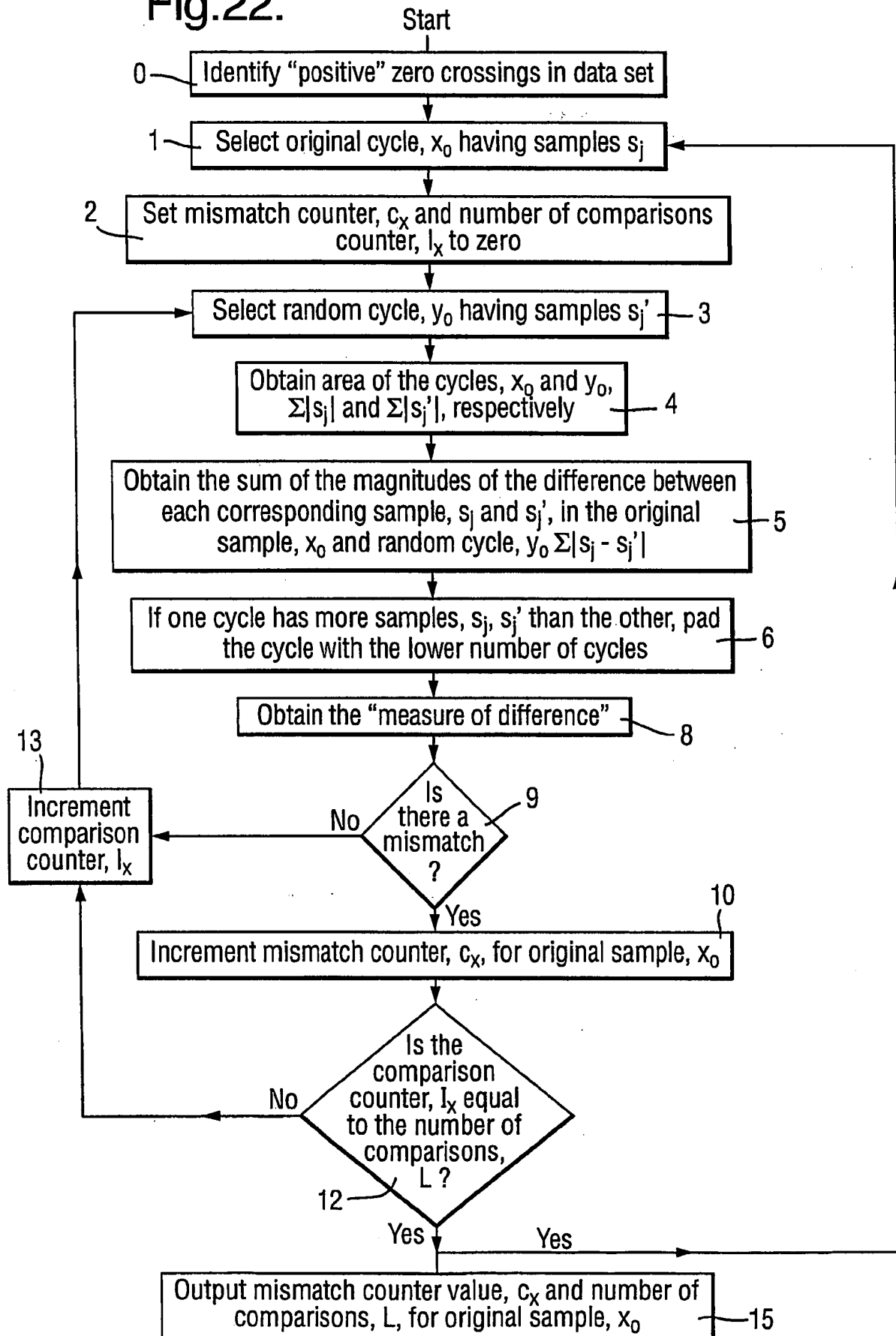


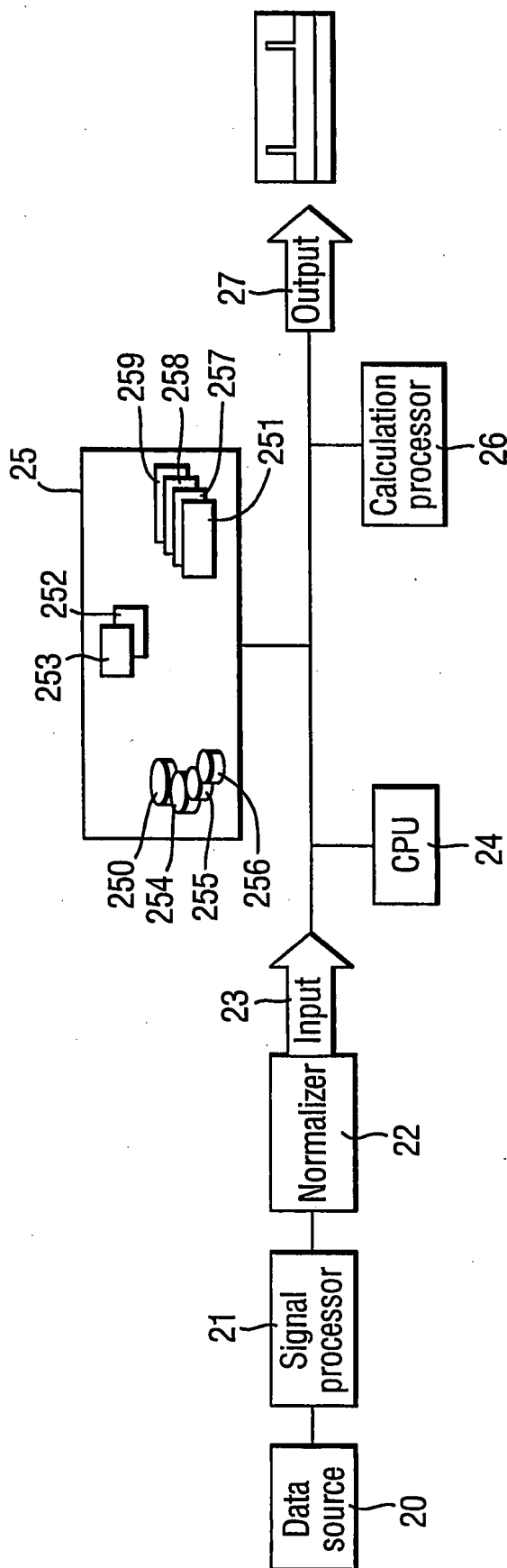
Fig.22.

17/26



18/26

Fig.23.



19/26

Fig.24.

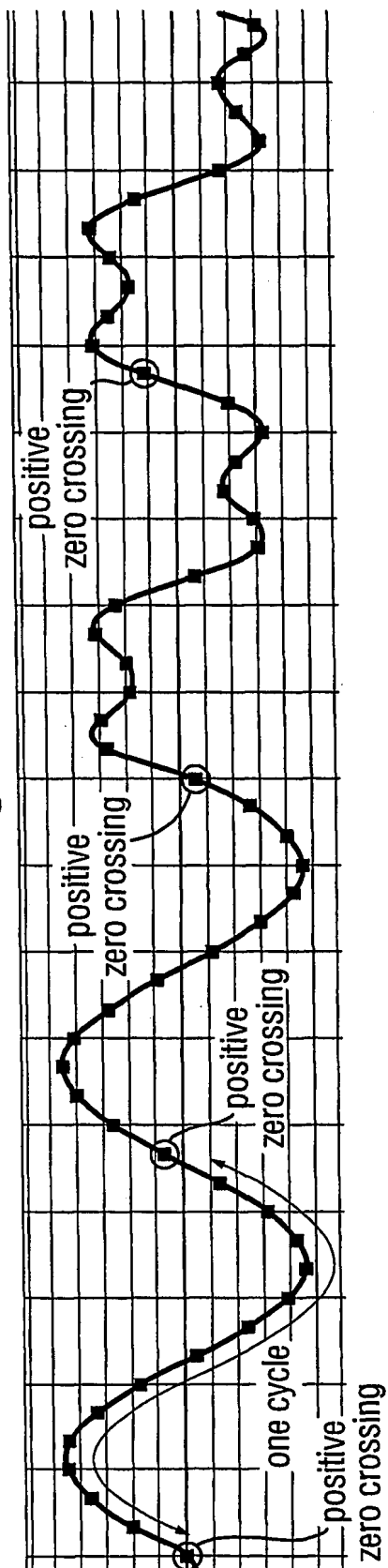
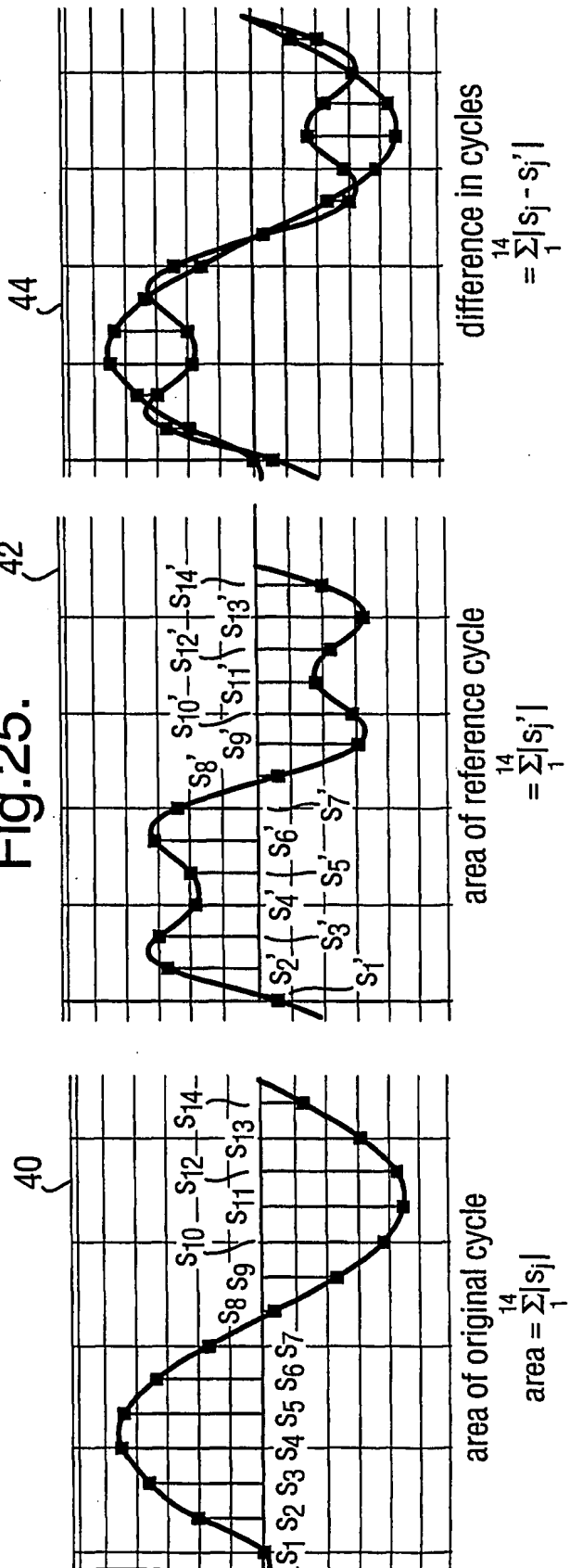


Fig.25.



20/26

Fig.26.

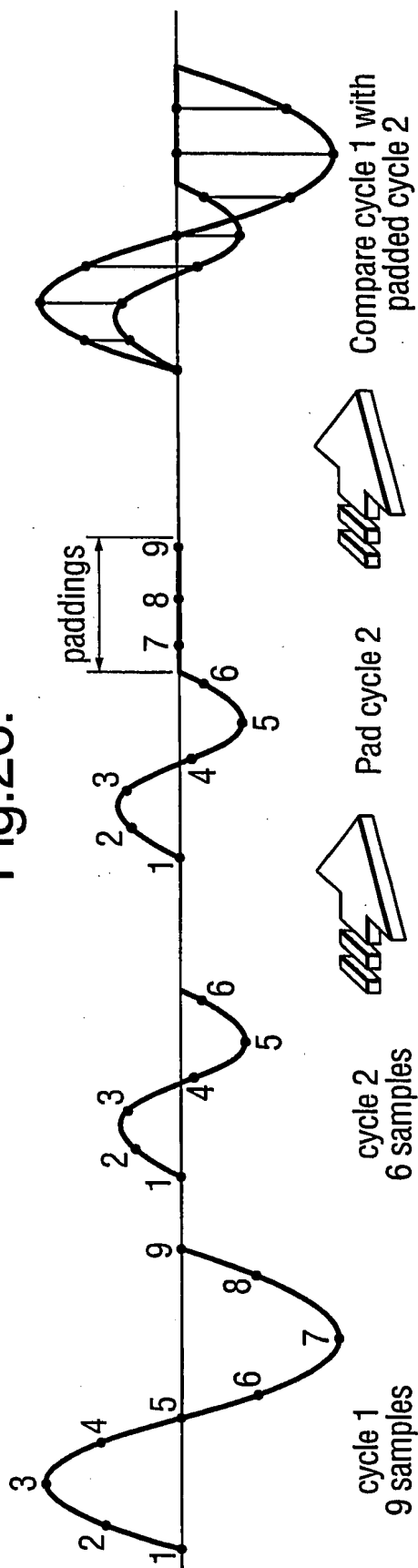


Fig.27.

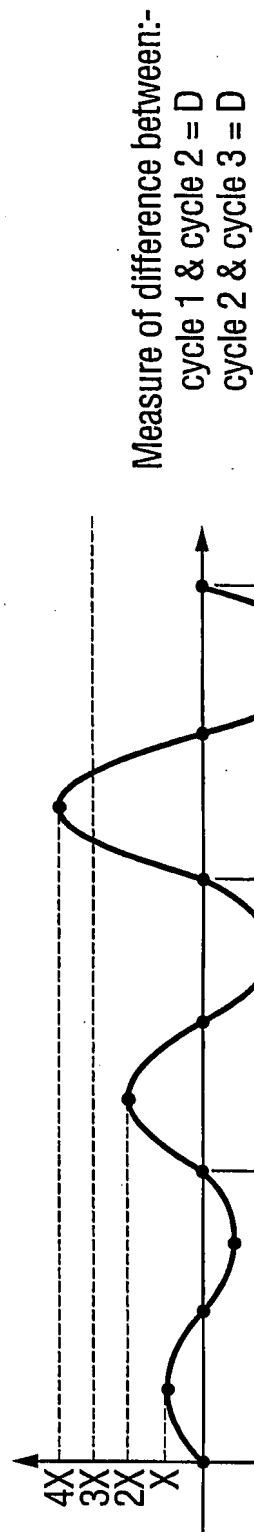


Fig.28.

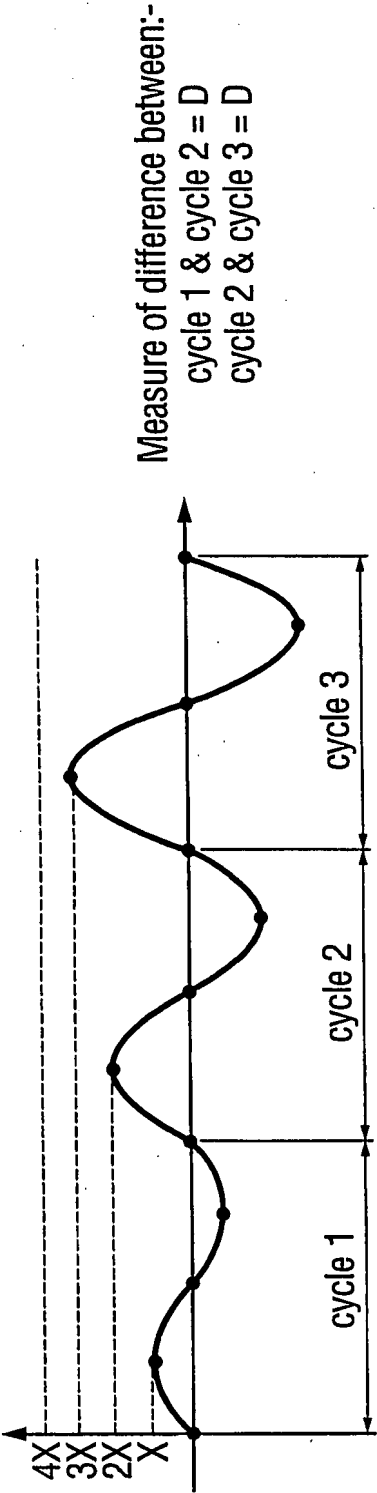


Fig.29.

Result 1

Binary sequence of 0's and 1's. 500 elements in total		250 comparisons, threshold = 0.1 (not critical)	
Perfect discrimination of anomalous cycles			

22/26

Fig.30.

Result 2

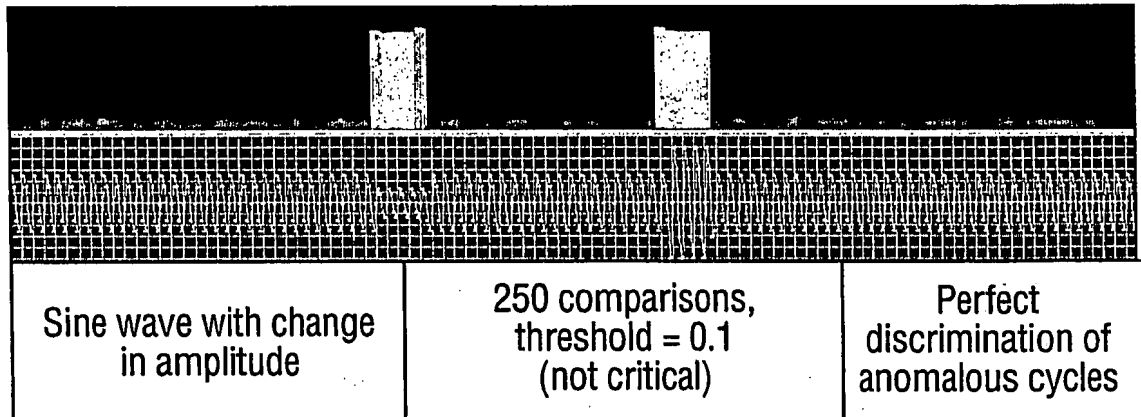


Fig.31.

Result 3

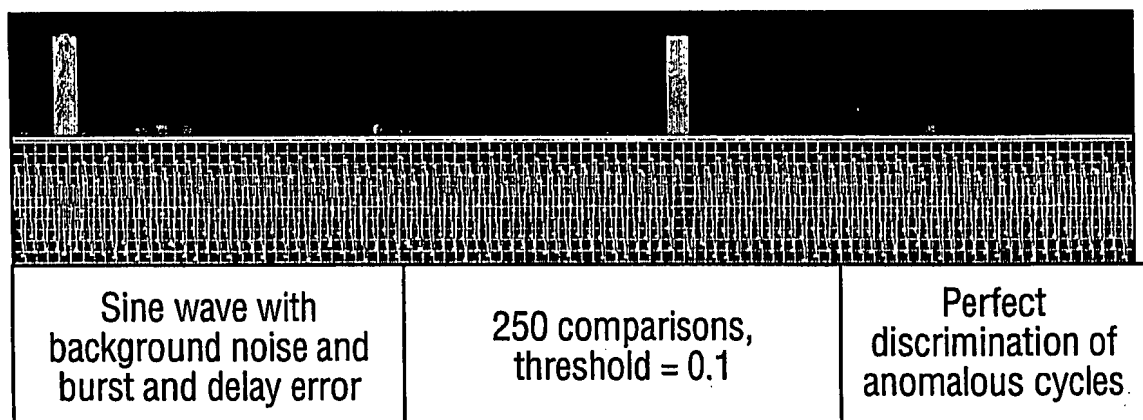
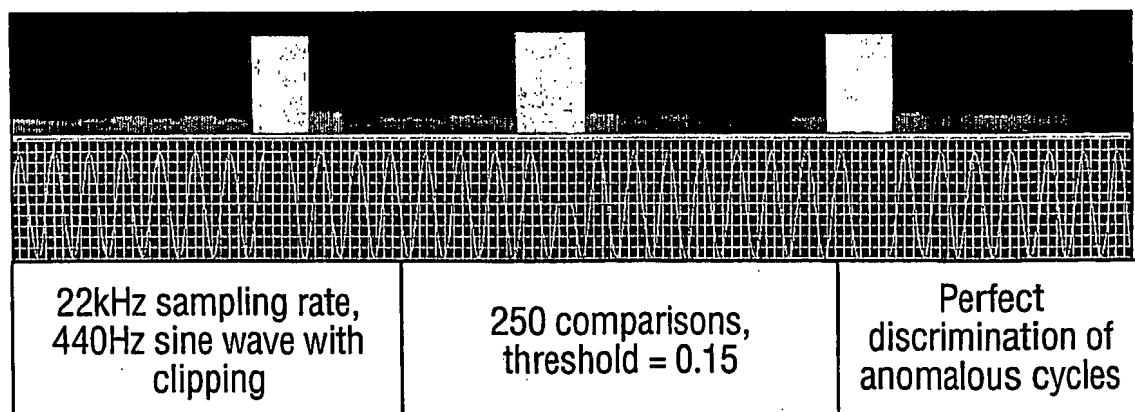


Fig.32.

Result 4



23/26

Fig.33.

Result 5

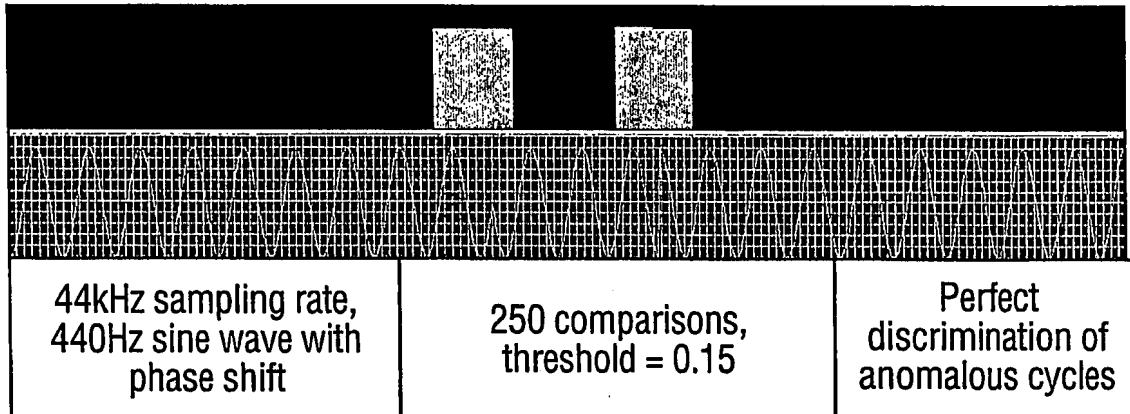
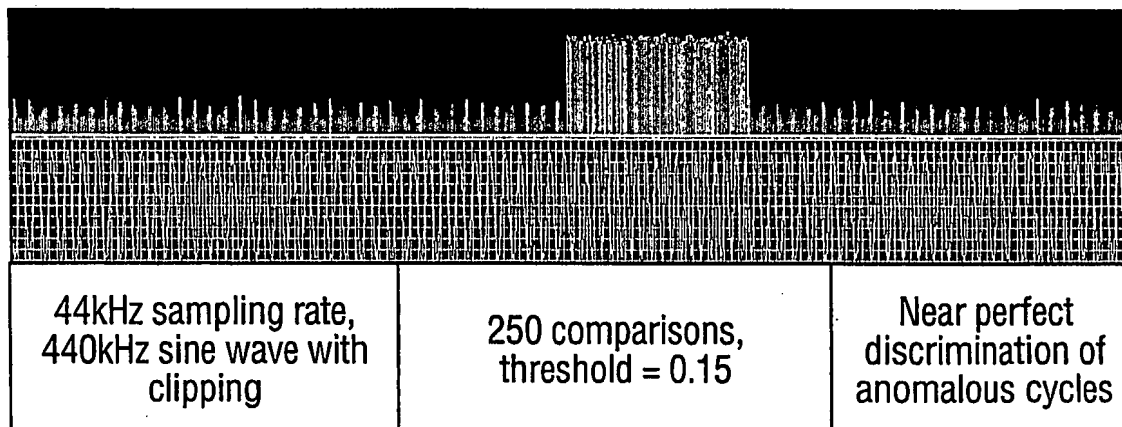


Fig.34.

Result 6



BEST AVAILABLE COPY

24/26

Fig.35.

Result 7

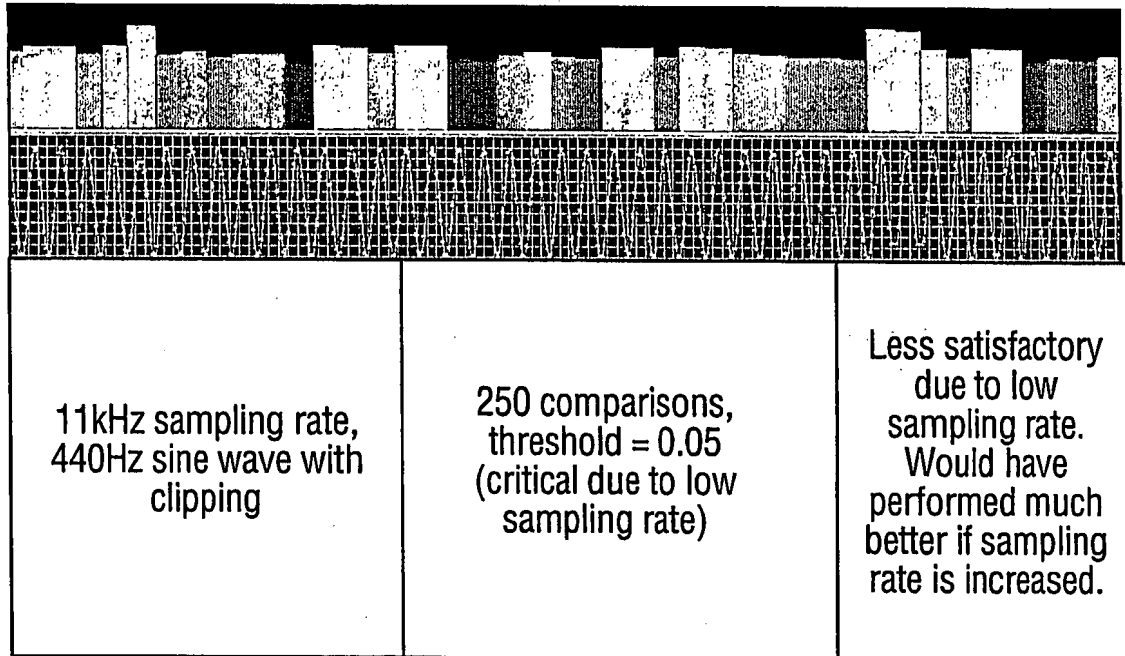
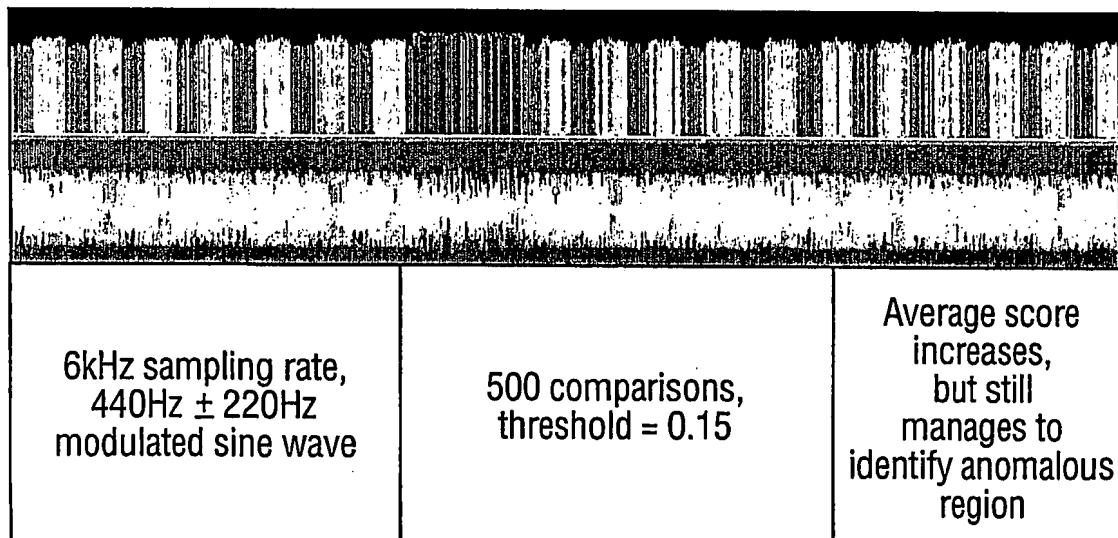


Fig.36.

Result 8



BEST AVAILABLE COPY

25/26

Fig.37.

Result 9

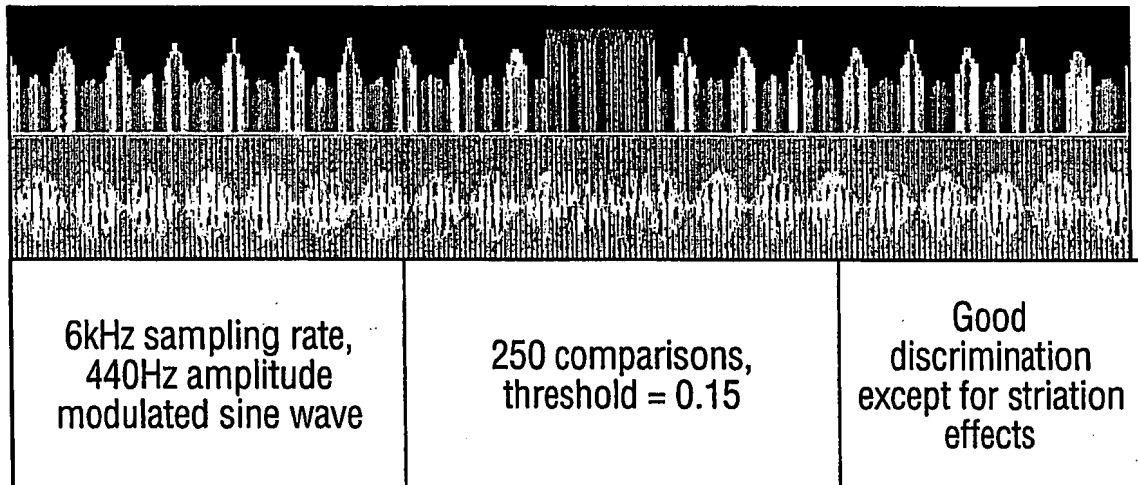
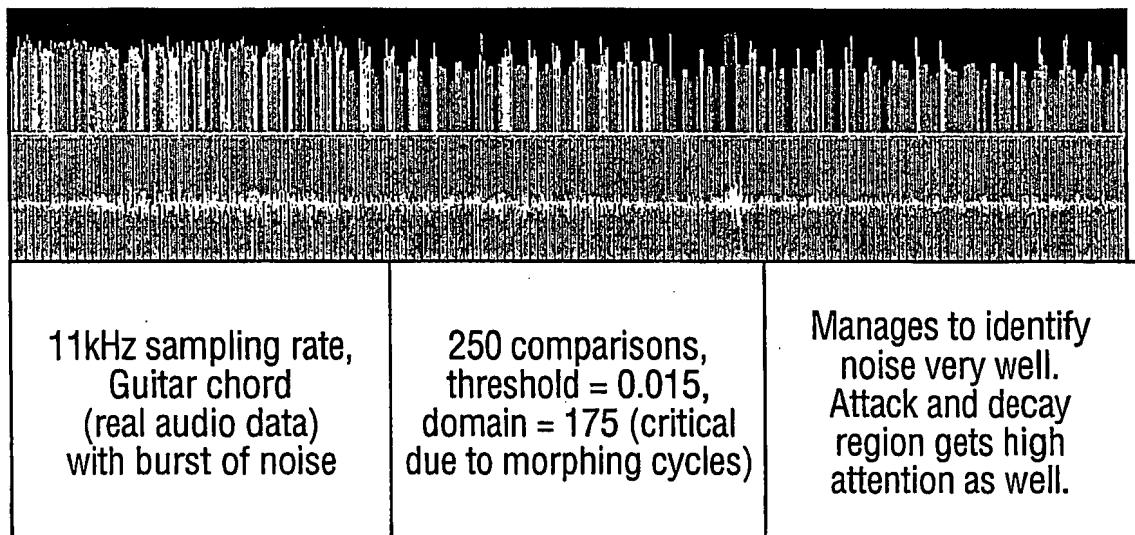


Fig.38.

Result 10



26/26

Fig.39.

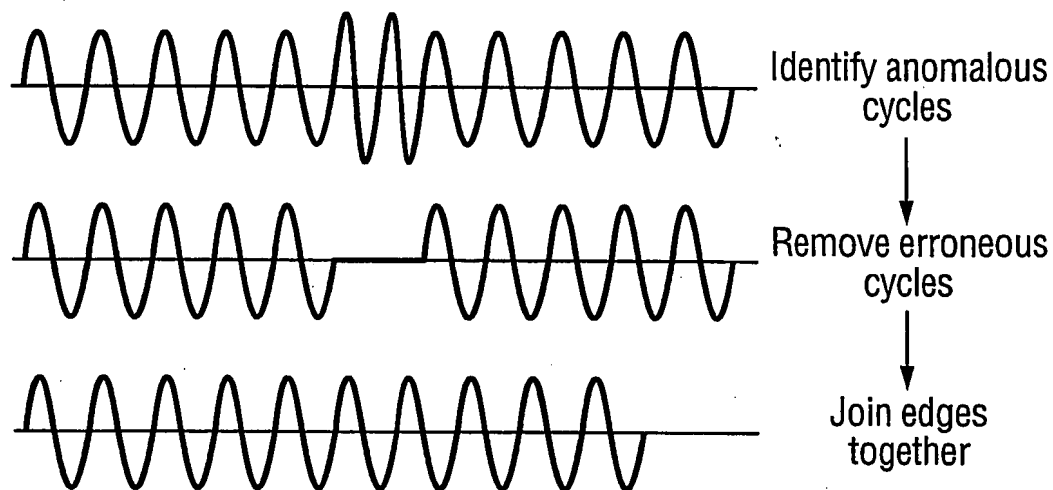
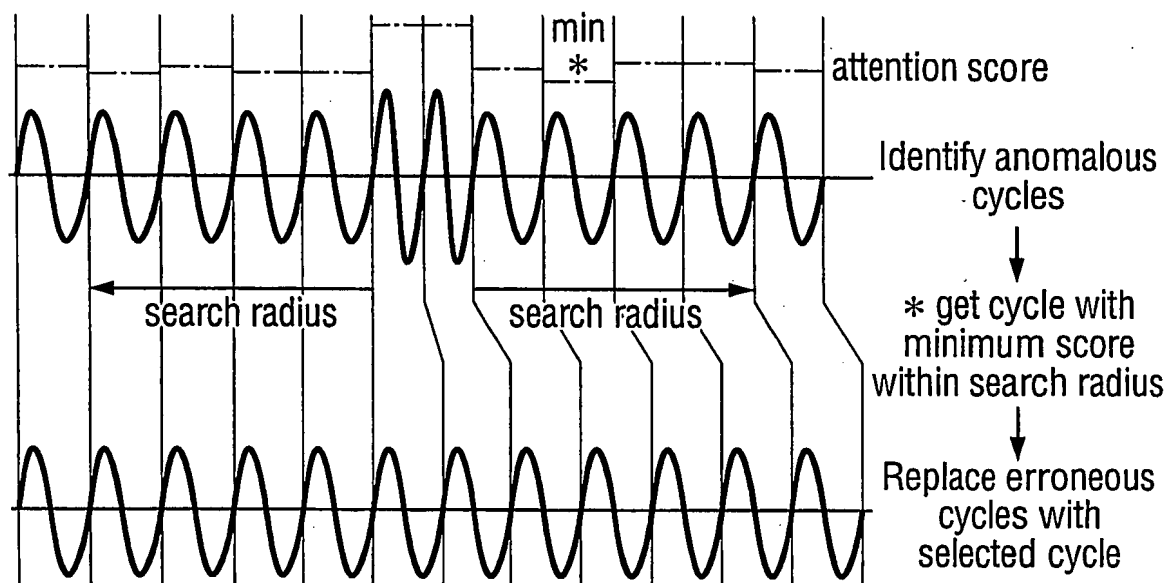


Fig.40.



BEST AVAILABLE COPY

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 03/01211

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G10L19/00 H03M7/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G10L H03M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the International search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 02 21446 A (BRITISH TELECOMM ;STENTIFORD FREDERICK WARWICK M (GB)) 14 March 2002 (2002-03-14)	1-9, 14-23
Y	the whole document	10-13
X	WO 01 31638 A (ERICSSON TELEFON AB L M) 3 May 2001 (2001-05-03) the whole document	1,20-23
X	US 5 867 813 A (ZGLINSKI ZIEMOWIT ET AL) 2 February 1999 (1999-02-02) the whole document	1,20-23
	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the International filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the International filing date but later than the priority date claimed

- *T* later document published after the International filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *Z* document member of the same patent family

Date of the actual completion of the International search

12 June 2003

Date of mailing of the International search report

24/06/2003

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Toumpoulidis, T

INTERNATIONAL SEARCH REPORT

International Application No

PCT/GB 03/01211

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	<p>ZHAO D ET AL: "Morphology on detection of calcifications in mammograms" DIGITAL SIGNAL PROCESSING 2, ESTIMATION, VLSI. SAN FRANCISCO, MAR. 23 - 26, 1992, PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), NEW YORK, IEEE, US, vol. 5 CONF. 17, 23 March 1992 (1992-03-23), pages 129-132, XP010059006 ISBN: 0-7803-0532-9 the whole document</p> <p>-----</p>	10-13

INTERNATIONAL SEARCH REPORT

tion on patent family members

International Application No

PCT/GB 03/01211

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 0221446	A	14-03-2002	AU 3202901 A	27-08-2001
			AU 8417101 A	22-03-2002
			CA 2400085 A1	23-08-2001
			EP 1281164 A2	05-02-2003
			EP 1316067 A1	04-06-2003
			WO 0161648 A2	23-08-2001
			WO 0221446 A1	14-03-2002
			US 2002081033 A1	27-06-2002
WO 0131638	A	03-05-2001	US 6499009 B1	24-12-2002
			AU 1319801 A	08-05-2001
			EP 1226577 A1	31-07-2002
			WO 0131638 A1	03-05-2001
US 5867813	A	02-02-1999	CA 2148340 A1	02-11-1996